



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

**INGENIERÍA EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES**

PROYECTO DE INVESTIGACIÓN

**“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE
DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE
COTOPAXI”**

**Proyecto de Investigación presentado previo a la obtención del Título de
Ingeniero en Informática y Sistemas Computacionales**

Autor:

Wilmer David Muyulema Andrade

Tutor:

Ing. M.Sc. Edel Angel Rodríguez Sánchez


La Maná - Ecuador

Octubre- 2017

DECLARACIÓN DE AUTORÍA

Yo, **MUYULEMA ANDRADE WILMER DAVID**, declaro ser autor del presente proyecto de investigación. **“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI”**, siendo el Ing.MSc. Edel Ángel Rodríguez Sánchez tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

.....

Wilmer David Muyulema Andrade
C.I. 1205974304

UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
LA MANÁ – ECUADOR

AVAL DEL DIRECTOR DE PROYECTO INVESTIGATIVO

En calidad de Tutor del Trabajo de Investigación sobre el título: **SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI**, de Wilmer David Muyulema Andrade, de la carrera de Ingeniería en Informática y Sistemas Computacionales, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Honorable Consejo Académico de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

La Maná, Octubre 2017


Ing. M.Sc. Edel Angel Rodríguez Sánchez
TUTOR DEL PROYECTO

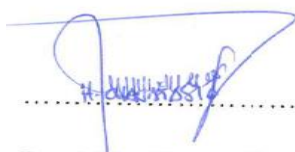
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la Facultad de Ciencias de la Ingeniería y Aplicadas; por cuanto, el postulante: WILMER DAVID MUYULEMA ANDRADE con el Título de Proyecto de Investigación: **SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI**, han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto.

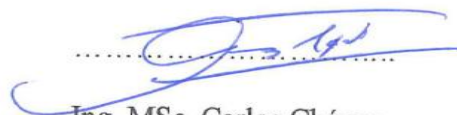
Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

La Maná, 31 de octubre 2017


Para constancia firman;



Ing. MSc. Henry Chanatasig
LECTOR 1 PRESIDENTE



Ing. MSc. Carlos Chávez
LECTOR 2



Ing. MSc. Johnny Bajaña
LECTOR 3

DEDICATORIA

Dedico este trabajo principalmente a Dios y a mis Padres, Gabriel & María por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación personal.

A mi querida esposa Daniela Ortiz y mi amada hija Leslie Valentina por ser los pilares más importantes y por demostrarme siempre su cariño y amor incondicional sin importar nuestras diferencia de opiniones.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

A mi Tutor Ing. MSc. Edel Rodríguez, gracias por su tiempo, por su apoyo así como la sabiduría que me transmitió en el desarrollo de formación personal.

David Muyulema Andrade

AGRADECIMIENTO

Agradezco a Dios por protegerme todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de la vida.

A mis padres, que con su demostración de padres ejemplares me han enseñado a no desfallecer ni rendirme ante nada y siempre a perseverar a través de sus sabios consejos.

A mi amada esposa, Daniela Ortiz por su apoyo incondicional y por demostrarme la gran fe que tiene en mí.

A mí querida hija, Leslie Valentina quien fue mi fuerza, voluntad y las ganas de superarme en mi vida profesional.

A mis herman@s, que con sus consejos me ha ayudado afrontar los retos que se me han presentado a lo largo de mi vida.

A mis compañer@s, de trabajo que de una u otra manera me apoyaron de una forma incondicional.

Al Ing. MSc. Edel Rodríguez, por toda la colaboración brindada, durante la elaboración de este proyecto.

David Muyulema Andrade

Índice

| | | |
|-----------|--|----------|
| 1. | TÍTULO DEL PROYECTO | 1 |
| 2. | DESCRIPCIÓN DEL PROYECTO | 2 |
| 3. | RESUMEN | 3 |
| 4. | ABSTRACT..... | 4 |
| 5. | BENEFICIARIOS DEL PROYECTO | 5 |
| 6. | PROBLEMA DE INVESTIGACIÓN | 6 |
| 7. | OBJETIVOS..... | 7 |
| 7.1. | OBJETIVO GENERAL | 7 |
| 7.2. | OBJETIVOS ESPECÍFICOS | 7 |
| 8. | ACTIVIDADES Y SISTEMA DE TAREAS EN RELACIÓN A LOS OBJETIVOS PLANTEADOS: | 8 |
| 9. | FUNDAMENTACIÓN CIENTÍFICO TÉCNICA..... | 9 |
| 9.1. | REQUERIMIENTOS FUNCIONALES | 9 |
| 9.2. | METODOLOGÍAS DE DESARROLLO DE SOFTWARE | 10 |
| 9.2.1. | Ingeniería de Software..... | 11 |
| 9.2.2. | Revisión de metodologías | 11 |
| 9.2.3. | Programación Extrema (Extreme Programming, Xp)..... | 13 |
| 9.2.4. | Evaluación y selección de la metodología | 14 |
| 9.2.5. | Las Historias de Usuario | 14 |
| 9.3. | PARADIGMA DE PROGRAMACIÓN | 14 |
| 9.3.1. | Programación Estructurada..... | 15 |
| 9.3.2. | Programación Orientada a Objetos | 16 |
| 9.3.3. | Selección..... | 17 |
| 9.4. | TECNOLOGÍAS WEB | 17 |
| 9.4.1. | Servidor WEB | 18 |
| 9.4.2. | Fundamentos de HTML..... | 19 |
| 9.4.3. | CSS | 20 |
| 9.5. | JAVASCRIPT | 22 |
| 9.5.1. | Especificaciones oficiales | 22 |
| 9.6. | PHP: HYPERTEXT PREPROCESSOR..... | 23 |
| 9.6.1. | Definiciones fundamentales de PHP | 23 |
| 9.6.2. | Código PHP en una página WEB HTML..... | 25 |
| 9.6.3. | ASP.NET | 26 |
| 9.6.4. | Marco de trabajo de ASP.Net | 26 |

| | | |
|------------|---|-----------|
| 9.7. | SISTEMAS DE GESTIÓN DE BASES DE DATOS | 28 |
| 9.7.1. | MySQL | 29 |
| 9.7.2. | Microsoft SQL Server | 31 |
| 9.7.3. | Ventajas de Microsoft SQL Server | 33 |
| 9.8. | FRAMEWORKS PARA EL DESARROLLO WEB..... | 33 |
| 9.8.1. | Symfony 2..... | 34 |
| 9.8.2. | NetBeans | 36 |
| 9.8.3. | Microsoft Visual Studio..... | 38 |
| 9.8.4. | Resultado del proceso de selección del las herramientas para el desarrollo..... | 41 |
| 9.9. | HERRAMIENTAS AUXILIARES PARA DE DESARROLLO WEB | 41 |
| 9.9.1. | Sublime Text | 41 |
| 9.9.2. | WampServer | 42 |
| 10. | HIPÓTESIS | 44 |
| 11. | METODOLOGÍA | 44 |
| 11.1. | TIPOS DE INVESTIGACIÓN | 44 |
| 11.1.1. | Investigación Bibliográfica | 44 |
| 11.2. | INVESTIGACIÓN EXPLORATORIA | 44 |
| 11.3. | MÉTODOS DE INVESTIGACIÓN | 45 |
| 11.3.1. | Método inductivo..... | 45 |
| 11.3.2. | Método deductivo | 45 |
| 11.3.3. | Método analítico | 46 |
| 11.4. | TÉCNICAS DE INVESTIGACIÓN | 46 |
| 11.4.1. | Observación directa | 46 |
| 12. | PRESUPUESTO PARA LA ELABORACIÓN DEL PROYECTO | 47 |
| 13. | ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS | 48 |
| 13.1. | MUESTREO ALEATORIO DE LA POBLACIÓN..... | 48 |
| 13.2. | ANÁLISIS DE LOS RESULTADOS DE LAS ENCUESTAS..... | 49 |
| 13.2.1. | Resultados de las encuestas aplicadas | 49 |
| 13.3. | REQUERIMIENTOS FUNCIONALES..... | 50 |
| 13.3.1. | Requerimientos funcionales de proceso o área de negocio..... | 50 |
| 13.3.2. | Requerimientos funcionales de interfaz gráfica | 51 |
| 13.3.3. | Requerimientos funcionales legales o regulatorios | 51 |
| 13.4. | REQUERIMIENTOS NO FUNCIONALES..... | 51 |
| 13.4.1. | Requerimientos de Eficiencia..... | 51 |
| 13.4.2. | Requerimientos de Seguridad lógica y de datos..... | 52 |
| 13.4.3. | Requerimientos de Usabilidad..... | 52 |
| 13.5. | SISTEMA EN FUNCIONAMIENTO | 53 |

| | | |
|------------|---|-----------|
| 13.5.1. | Pantalla principal | 53 |
| 13.5.2. | Gestión de roles | 54 |
| 13.5.3. | Gestión de usuarios..... | 55 |
| 13.5.4. | Gestión de indicadores de evaluación..... | 56 |
| 13.6. | PRUEBAS DE CAJA BLANCA Y CAJA NEGRA | 57 |
| 13.6.1. | Error de conversión de tipos incompatibles | 57 |
| 13.6.2. | Error de acceso a archivos ensamblados..... | 58 |
| 14. | CONCLUSIONES..... | 60 |
| 15. | RECOMENDACIONES | 60 |
| 16. | BIBLIOGRAFÍA..... | 61 |
| 17. | ANEXOS..... | 66 |
| 17.1. | ANEXO 1: MODELO RELACIONAL DE LA BASE DE DATOS | 66 |
| 17.2. | ANEXO 2: SCRIPT SQL DE LA BASE DE DATOS | 67 |
| 17.3. | ANEXO 3: ENCUESTA REALIZADA A LOS ESTUDIANTES | 69 |
| 17.4. | ANEXO 4: RESULTADOS DE LA ENCUESTA APLICADA A LOS ESTUDIANTES | 70 |
| 17.5. | ANEXO 4. ENCUESTA REALIZADA A LOS DOCENTES | 75 |
| 17.6. | ANEXO 5: RESULTADOS DE LA ENCUESTA APLICADA A LOS DOCENTES..... | 76 |

Índice de figuras

| | |
|---|----|
| Figura 1. Modelo Cliente – Servidor | 18 |
| Figura 2. Ejemplo de código HTML | 20 |
| Figura 3. Ejemplo de código CSS | 21 |
| Figura 4. Ejemplo de código JavaScript..... | 23 |
| Figura 5. Ejemplo de código PHP | 24 |
| Figura 6. Código PHP insertado en el código HTML | 25 |
| Figura 7. Código PHP insertado en el código HTML | 27 |
| Figura 8. Ejemplo del Modelo relacional | 29 |
| Figura 9. Arquitectura funcional de MySQL..... | 31 |
| Figura 10. Arquitectura funcional de MySQL..... | 32 |
| Figura 11. Organización del código para un proyecto de un blog en Symfony | 36 |
| Figura 12. Interfaz principal de NetBeans..... | 38 |
| Figura 13. Interfaz principal del software Visual Studio..... | 40 |
| Figura 14. Notepad++..... | 42 |
| Figura 15. Pantalla principal..... | 53 |
| Figura 16. Menú principal del administrador | 53 |
| Figura 17. Gestión de roles..... | 54 |
| Figura 18. Nuevo role | 54 |
| Figura 19. Gestión de usuarios | 55 |
| Figura 20. Lista de indicadores evaluativos | 56 |
| Figura 21. Gestión de indicadores evaluativos | 57 |
| Figura 22. Error de conversión de tipos | 58 |
| Figura 23. Error de acceso a archivos ensamblado | 58 |

Índice de tablas

| | |
|---|----|
| Tabla 1. Beneficiarios directos e indirectos..... | 5 |
| Tabla 2. Actividades | 8 |
| Tabla 3. Presupuesto..... | 47 |

1. Título del proyecto

“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LAS EVALUACIONES DE DESEMPEÑO EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

Fecha de inicio: 1 de Mayo del 2017

Fecha de finalización: Octubre del 2017

Lugar de ejecución: Universidad Técnica de Cotopaxi Extensión La Maná

Facultad que auspicia: Facultad de Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia: Ingeniería en informática y Sistemas Computacionales

Proyecto de investigación vinculado: Universidad Técnica de Cotopaxi

Equipo de trabajo:

Nombres: Wilmer David Muyulema Andrade

Correo: dmuyulemaandrade@gmail.com

Teléfono: 0959156255

Tutor: Ing. M.Sc. Edel Ángel Rodríguez Sánchez

Correo: edel.rodriguez@utc.edu.ec

Teléfono: 0989930089

Área de Conocimiento: Desarrollo de Software

Línea de investigación: Línea 6: Tecnología de la Información y Comunicación (TICS) y diseño Gráfico.

Sub líneas de investigación de la Carrera: Ingeniería de Software

Categorización: Desarrollo de software

2. Descripción del proyecto

En esta investigación se propone el desarrollo de un sistema informático que permita mediante la Web la gestión de las evaluaciones del desempeño de docentes en la Universidad Técnica de Cotopaxi.

Con la realización de este sistema se pretende favorecer el proceso de gestión de las evaluaciones de desempeño. Para tal fin se propone realizar una aplicación que por medio de la Web permita a los directivos de la institución gestionar las evaluaciones de desempeño de los docentes de la institución.

El sistema mediante una base de datos y un lenguaje de programación de alto nivel, implementará una interfaz amigable al usuario facilitándole a este todo lo referente a la gestión de las evaluaciones de desempeño.

Para conseguir que el sistema sea insertado sin dificultades en el sitio oficial de la institución, el mismo será implementado sobre la plataforma .NET por medio de Visual Studio y se usará para almacenar la información Microsoft SQL Server como gestor de la base de datos.

En el presente proyecto dentro del campo de investigación se utilizarán la técnica de entrevista dirigida al coordinador encargado en la matriz así como también al personal que labora dentro de la misma en el área determinada a trabajar. De esta manera se obtendrá información útil la cual será utilizada para llegar a la solución del problema planteado.

Palabras claves: Desarrollo, Programación, Ingeniería de Software, Implementación, Visual Studio, SQL Server, Evaluación del Desempeño Docente.

3. Resumen

La presente Investigación tiene lugar debido a la solicitud por parte de los directivos de la Universidad Técnica de Cotopaxi (UTC) de un sistema que permita a los estudiantes y coordinadores de carrera manejar la información referente a las evaluaciones de los docentes de esta institución. Además, se evidencia que actualmente no se dispone de una herramienta que permita realizar esta actividad.

Esta investigación pretende aportar un sistema informático para la gestión de las evaluaciones de desempeño de los docentes en la Universidad Técnica de Cotopaxi. A su vez, dicho sistema permitirá que los estudiantes evalúen por medio de internet a los docentes que les han impartido clases. El sistema permitirá además la evaluación de los docentes por parte de los coordinadores de carrera y pares académicos, llevando el registro histórico de esta información para ser consultada por los directivos en cualquier momento.

Con la instalación de este sistema informático para la gestión de información se beneficia primeramente a los directivos de la Universidad Técnica de Cotopaxi. Por otro lado, se ven beneficiados los estudiantes de esta institución.

El sistema propuesto tiene impacto directo en el proceso de evaluación de desempeño de los docentes en la Universidad Técnica de Cotopaxi. En la práctica el sistema tiene como utilidad la agilización de este proceso permitiendo que los estudiantes evalúen de forma anónima, esto significa que los docentes no conocerán la evaluación que obtuvieron de sus estudiantes. Esta información sólo podrá ser consultada por los directivos de la institución o el administrador del sistema.

4. Abstract

The present research is due to the request which was sent by the management of the Technical University of Cotopaxi (UTC) of a system that allows students and coordinators of the career to manage the information about assessment of the professors who work at the university. Besides, nowadays the university doesn't have this tool which allow to do this activity.

This research aims to contribute a computer system for the assessment of the professors' performance at Cotopaxi Technical University. So the system will allow students to evaluate them through the Internet. The system will also allow of the career to evaluate their professors obtaining the register of the information to be analyzed by the management at any moment.

With installation of this computer system for the management of the information. The management and students of Cotopaxi Technical University will be benefit.

The system has a direct impact on the process of performance assessment of the professor's Cotopaxi Technical University. In the practice, the system has the advantage of streamlining this process allowing students to evaluate anonymously, this means that professors won't know the result that they get from their students. This information could be consulted by management of Cotopaxi Technical University or the systems administrator.

5. Beneficiarios del proyecto

Con el desarrollo del presente sistema se pretende, en principio, beneficiar a los directivos de la institución relacionados de alguna manera con el proceso de evaluación del desempeño e indirectamente se pretende beneficiar a los estudiantes y docentes; los primeros dispondrán de un método ágil y seguro para realizar esta tarea, mientras que los segundos tendrán un resultado confiable sobre el cual trabajar para mejorar sus actividades académicas y educativas.

| BENEFICIADOS | CANTIDAD |
|---------------------------------|-----------------|
| BENEFICIARIOS DIRECTOS | |
| Docentes | 360 |
| BENEFICIARIOS INDIRECTOS | |
| Estudiantes | 9000 |
| TOTAL | 9360 |

Tabla 1. Beneficiarios directos e indirectos

Fuente: Secretaria Académica. Universidad Técnica de Cotopaxi

6. Problema de investigación

La evaluación de desempeño del docente, a nivel internacional, es un proceso permanente que permite verificar el quehacer didáctico y profesional de los educadores identificando en distintas partes del mundo, además permite identificar sus fortalezas y aspectos a mejorar. A través de la valoración de sus competencias se identifican los puntos a trabajar, los cuales serán una pieza clave para la construcción de planes de mejoramiento institucional y desarrollo personal y profesional de docentes. (Veloz, 2010)

En nuestro país todas las universidades, públicas o privadas, tienen preocupaciones sobre algunos contenidos de la Ley de Educación Superior, o sobre ciertas disposiciones o procedimientos de su función o su evaluación. El Consejo de Educación Superior (CES), la Secretaría Nacional de Educación Superior, Ciencia y Tecnología (Senescyt) y el Consejo de Evaluación, Acreditación y de Aseguramiento de la Calidad de la Educación Superior (Ceaaces) definen la evaluación del desempeño docente como el conjunto de acciones organizadas de acuerdo con las mediaciones e interacciones pedagógicas entre el conocimiento científico y el conocimiento formativo de los estudiantes universitarios, y con las mediaciones socioculturales y lingüísticas. Además estos organismos también exponen que la evaluación del desempeño docente permitirá promover acciones didáctico-pedagógicas que favorezcan los procesos de aprendizaje de los estudiantes, y el mejoramiento de la formación inicial docente, así como su desarrollo profesional y que la evaluación del docente no debe verse como un acto fiscalizador, sino como una forma de fomentar y favorecer su perfeccionamiento.

La Universidad Técnica de Cotopaxi actualmente se encuentra en un proceso de mejoramiento de los procesos docentes y además dedica esfuerzo y recursos por lograr la mejor calidad de los graduados universitarios, es por esto que se ha diseñado un nuevo modelo de evaluación a los docentes con el fin de garantizar la mejor calidad de estos y a la vez, favorecer la calidad de sus graduados. Teniendo en cuenta los adelantos actuales de las Tecnologías de la Informática y las Comunicaciones (TIC) se implementará un sistema informático que permita realizar de forma eficiente y sencilla este proceso de evaluación. Se pretende que este sistema a través de la WEB permita realizar este proceso de evaluación por parte de los estudiantes y de los coordinadores de carrera para los docentes de la institución.

7. Objetivos

7.1. Objetivo general

- Favorecer la gestión de la información referente a las evaluaciones del desempeño docente en la Universidad Técnica de Cotopaxi mediante un sistema informático que permita el seguimiento mediante la Web de los resultados evaluados en cada docente.

7.2. Objetivos específicos

- Capturar los requerimientos funcionales para el sistema por medio de entrevistas con los directivos de la Entidad.
- Analizar de las herramientas disponibles en la actualidad para el desarrollo de sistemas informáticos.
- Desarrollar el sistema informático aplicando las herramientas adecuadas para conseguir un desarrollo eficiente cumpliendo los plazos de entrega.

Tabla 2. Actividades

| 8. Actividades y sistema de tareas en relación a los objetivos planteados: | | | |
|--|---|--|---|
| OBJETIVOS | ACTIVIDADES | RESULTADO | DESCRIPCIÓN |
| Capturar los requerimientos funcionales para el sistema por medio de entrevistas con los directivos de la Entidad | Entrevista con los directivos de la institución. Modelado del proceso actual de evaluación de los docentes. Elaboración de los requerimientos solicitados por los directivos | Listado de requerimientos funciones y no funcionales del sistema agrupados según su clasificación en la Ingeniería de Requerimientos | Entrevista presencial. Modelado UML. Análisis de procesos de gestión. Aplicación de métodos de la Ingeniería de requerimientos |
| Analizar de las herramientas disponibles en la actualidad para el desarrollo de sistemas informáticos | Consultar de las herramientas de desarrollo de software más usadas en la actualidad. Comparar las herramientas consultadas para determinar la que mejor se adecua a la satisfacción de los requerimientos funcionales. | ToolBox con las herramientas adecuadas para el desarrollo | Consultas en Internet Descarga y ejecución de las herramientas Comprobación del rendimiento del código generado |
| Desarrollar el sistema informático aplicando las herramientas adecuadas para conseguir un desarrollo eficiente cumpliendo los plazos de entrega | Aplicación de las herramientas seleccionada sobre los requerimientos funcionales. | Aplicación compilada lista para ser desplegada en la WEB | Diseño de la base de datos Crear las interfaces del sistema Escritura del código fuente Pruebas del sistema desarrollado |

9. Fundamentación Científico Técnica

9.1. Requerimientos funcionales

Los requerimientos funcionales definen una determinada función deseada para el sistema informático o alguno de sus componentes. Se trata de una función descrita por el cliente como un conjunto de entradas, comportamientos y salidas. Los requerimientos ó como también suelen llamarse, requisitos funcionales, pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone que el sistema objeto de desarrollo debe cumplir. Los requisitos de comportamiento para cada requisito funcional se muestran en los casos de usos de la metodología RUP, sobre la que se tratará más adelante y son complementados por los requerimientos no funcionales, que se enfocan en cambio en la implementación o el diseño.

De análisis del proceso actual de evaluación de desempeño en la institución se identificaron como requerimientos funcionales los siguientes puntos:

- Se permitirá el registro de nuevos estudiantes y docentes los cuales podrán modificar la información de su perfil una vez ingresados al sistema.
- Al realizar una evaluación por parte de alguna entidad, ésta estará disponible para ser consultada por los ejecutivos superiores.
- El sistema permitirá a los usuarios autorizados el ingresar y/o modificar los aspectos a ser evaluados en cada entidad.
- El sistema permitirá aprobar, cambiar o actualizar aspectos de evaluación.
- A cada orden se le registro de evaluación se le asignará un identificador único, que será utilizado para identificarla en todos los procesos subsecuentes que se realicen sobre esta.
- Al ingresar un registro de evaluación este estará asociado a un docente específico.
- El sistema permitirá la realización de varios planes de evaluación con tal de asignar estos a docentes o a coordinadores de carrera, tal como lo indiquen las autoridades metodológicas

9.2. Metodologías de desarrollo de software.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. (Canós, Letelier, & Penadés, 2013)

Modelo secuencial. Representado por metodologías tan famosas como Waterfall. Se inicia con un completo análisis de los requisitos de los usuarios. En el siguiente paso, los programadores implementan el diseño y finalmente, el completado y perfecto sistema es probado y enviado.

Desarrollo incremental. Su principal objetivo es reducir el tiempo de desarrollo, dividiendo el proyecto en intervalos incrementales superpuestos. Del mismo modo que con el modelo waterfall, todos los requisitos se analizan antes de empezar a desarrollar, sin embargo, los requisitos se dividen en “incrementos” independientemente funcionales.

Desarrollo iterativo. A diferencia del modelo incremental se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos. En el desarrollo iterativo se rompe el proyecto en iteraciones de diferente longitud, cada una de ellas produciendo un producto completo y entregable.

Modelo en espiral. Comprende las mejores características de ciclo de vida clásico y el prototipado (desarrollo iterativo). Además, incluye el análisis de alternativas, identificación y reducción de riesgos (Gómez, López, & Bacalla, 2011)

9.2.1. Ingeniería de Software

La Ingeniería de Software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos. Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación. El ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto.

La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema (Pressman, 2004).

Cabe destacar que el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

La concepción fija el alcance del proyecto y desarrolla el modelo de negocio; la elaboración define el plan del proyecto, detalla las características y fundamenta la arquitectura; la construcción es el desarrollo del producto; y la transición es la transferencia del producto terminado a los usuarios.

Una vez que se completa este ciclo, entra en juego el mantenimiento del software. Se trata de una fase de esta ingeniería donde se solucionan los errores descubiertos (muchas veces advertidos por los propios usuarios) y se incorporan actualizaciones para hacer frente a los nuevos requisitos. El proceso de mantenimiento incorpora además nuevos desarrollos, para permitir que el software pueda cumplir con una mayor cantidad de tareas.(Newkirk & Martin, 2015)

9.2.2. Revisión de metodologías

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios de estos, cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se resumen dichas metodologías ágiles, dejando el análisis más detallado de XP para la siguiente sección.

SCRUM. Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.(Schwaber & Beedle, 2012)

Crystal Methodologies. Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). (Cockburn, 2016)

Dynamic Systems Development Method⁶ (DSDM). Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.(Stapleton, 2013)

Adaptive Software Development⁷ (ASD). Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La

revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.(Highsmith, 2013)

Feature-Driven Development® (FDD). Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.(Coad, Luca, & Lefebvre, 2011)

Lean Development® (LD). Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.(Poppendieck & Poppendieck, 2013)

9.2.3. Programación Extrema (Extreme Programming, Xp)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Letelier, 2016)

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas. (Letelier, 2016)

9.2.4. Evaluación y selección de la metodología

Una vez realizado el estudio de diferentes metodologías para el desarrollo de software, se obtuvieron los datos necesarios para la realizar un proceso de comparación que permitiera la selección de la metodología que mejor se adecuase a los requerimientos de nuestra investigación; siendo la metodología XP la que cumplió con estos requisitos. Seguidamente se describen los aspectos más importantes de esta metodología.

9.2.5. Las Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Jeffries, Anderson, & Hendrickson, 2011)

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, más quizás una estimación de esfuerzo en días. Beck en su libro presenta un ejemplo de ficha (customer story and task card) en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado cosas por terminar y comentarios.(Newkirk & Martin, 2015)

9.3. Paradigma de Programación

Un paradigma de programación indica un método de realizar cálculos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa. Los paradigmas fundamentales están asociados a determinados modelos de cómputo, también se asocian a un determinado estilo de programación. Por otra parte, los

lenguajes de programación suelen implementar, a menudo de forma parcial, varios paradigmas. (Joyanes Aguilar, 1996)

9.3.1. Programación Estructurada

La visión clásica de la programación estructurada se refiere al control de ejecución. El control de su ejecución es una de las cuestiones más importantes que hay que tener en cuenta al construir un programa en un lenguaje de alto nivel. La regla general es que las instrucciones se ejecuten sucesivamente una tras otra, pero diversas partes del programa se ejecutan o no dependiendo de que se cumpla alguna condición. Además, hay instrucciones (los bucles) que deben ejecutarse varias veces, ya sea en número fijo o hasta que se cumpla una condición determinada.

Sin embargo, algunos lenguajes de programación más antiguos (como Fortran) se apoyaban en una sola instrucción para modificar la secuencia de ejecución de las instrucciones mediante una transferencia incondicional de su control (con la instrucción goto, del inglés "go to", que significa "ir a"). Pero estas transferencias arbitrarias del control de ejecución hacen los programas muy poco legibles y difíciles de comprender. (Joyanes Aguilar, Fundamentos de programación, 2014)

A finales de los años sesenta, surgió una nueva forma de programar que reduce a la mínima expresión el uso de la instrucción goto y la sustituye por otras más comprensibles. Esta forma de programar se basa en un famoso teorema, desarrollado por Edsger Dijkstra, que demuestra que todo programa puede escribirse utilizando únicamente las tres estructuras básicas de control siguientes:

Secuencia: el bloque secuencial de instrucciones, instrucciones ejecutadas sucesivamente, una detrás de otra.

Selección: la instrucción condicional con doble alternativa, de la forma:

```
if (condición) then
    instrucción1
else
    instrucción2
```

Iteración: el bucle condicional "while condición do instrucción", que ejecuta la instrucción repetidamente mientras la condición se cumpla.

Los programas que utilizan sólo estas tres instrucciones de control básicas o sus variantes (como los bucles `for`, `repeat` o la instrucción condicional `switch-case`), pero no la instrucción `goto`, se llaman estructurados.

9.3.2. Programación Orientada a Objetos

Uno de los paradigmas de programación más importantes en la actualidad es la programación orientada a objetos (POO), la cual es una forma de programar que trata de encontrar una solución a los problemas que ocasionaba la programación estructurada. Esta introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Herencia: (por ejemplo, herencia de la clase D a la clase C) Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (`private`) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de OOP.

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio

en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Propiedad o atributo: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Estado interno: es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase. (Joyanes Aguilar, Programación orientada a objetos, 1996)

9.3.3. Selección

Atendiendo a los requerimientos de los sistemas informáticos en la actualidad y a los lenguajes empleados en el desarrollo de la investigación, entre los cuales figuran ASP.Net, PHP y JavaScript, donde estos integran entre sus mayores fortalezas, el modelo de programación orientado a objetos; luego de esto, se determina seleccionar el paradigma de POO como la base para la implementación de sistema informático objeto de este trabajo de esta investigación.

9.4. Tecnologías WEB

Fundación World Wide Web, que hace el seguimiento y análisis de la evolución de la Web. Mucha gente confunde (erróneamente) internet y la Web. Debes saber que la Web es parte de internet. Es un gran paquete que incluye, entre otros: la Web, correos

electrónicos, mensajería instantánea, etc. Tim Berners-Lee no es el inventor de internet, es "solo" inventor de la Web. (Consortium & others, 2011)

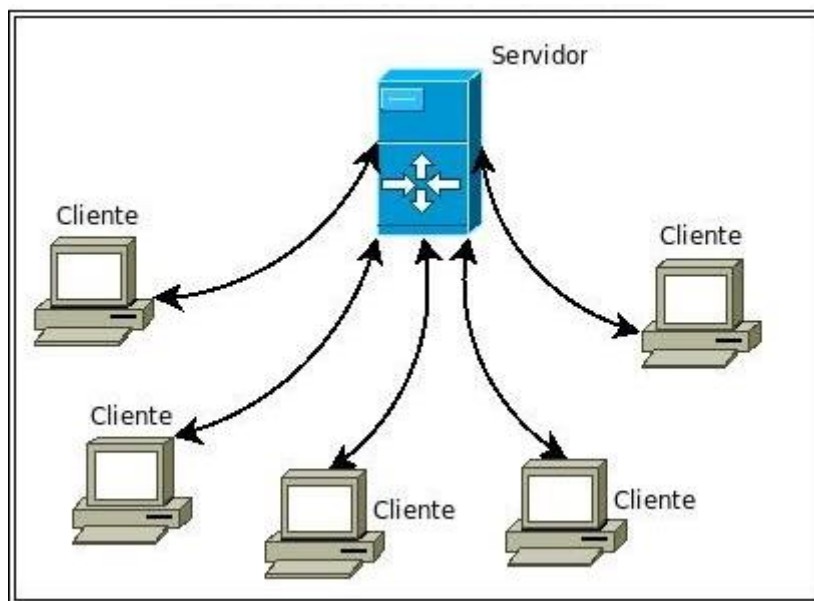


Figura 1. Modelo Cliente – Servidor

Fuente: Equipo de Investigación

HTML y CSS son la base del funcionamiento de casi todos los sitios web. Cuando visitas un sitio mediante un navegador, debes saber que, detrás de las escenas, las “ruedas” se activan para permitir que el sitio web se muestre. El ordenador se basa en lo que se indica en HTML y CSS para saber lo que ha de mostrar, la figura siguiente indica parte del código de la página mostrada previamente.

9.4.1. Servidor WEB

Un servidor Web es un programa que utiliza el protocolo de transferencia de hiper texto, HTTP (Hypertext Transfer Protocol), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras. Las computadoras y los dispositivos dedicados también pueden denominarse servidores Web.(Gauchat, 2012)

En un modelo cliente/servidor, todos los equipos que alojan sitios Web deben tener programas de servidor Web. Los principales servidores Web incluyen Apache (el servidor Web más ampliamente instalado), Internet Information Server (IIS) de Microsoft y nginx (que se pronuncia engine X) de NGNIX. Otros servidores Web

incluyen el servidor NetWare de Novell, el servidor Web de Google (GWS) y la familia de servidores Domino de IBM. (Agrawal & Gupta, 2014)

Los servidores Web a menudo forman parte de un paquete más amplio de programas relacionados con internet e intranet para servir correo electrónico, descargar solicitudes de archivos de protocolo de transferencia de archivos (FTP) y crear y publicar páginas Web. Las consideraciones al elegir un servidor Web incluyen cuán bien funciona con el sistema operativo y otros servidores, su capacidad para manejar la programación del servidor, las características de seguridad y las herramientas particulares de publicación, motor de búsqueda y creación de sitios que vienen con él. (Vértice, 2015)

9.4.2. Fundamentos de HTML

HTML es el lenguaje que se emplea para el desarrollo de páginas de internet. Las siglas de HTML significan HyperText Markup Lenguaje (lenguajes de marcas de Hypertexto). Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc. (Powell, 2011)

HTML es un lenguaje artificial que los ordenadores son capaces de interpretar y diseñado para que los programadores redacten instrucciones que los navegadores ejecutan para originar la página web. Es decir, HTML es un lenguaje de programación, o un "idioma que la máquina entiende y procesa para dar una respuesta". Un ejemplo de esto puede verse en la Figura 2. (Vértice, 2015)

Para simplificar se puede decir que el Hypertexto es aquel texto que pulsamos con el ratón del ordenador y nos conduce a otro texto cuando utilizamos internet. Pero además de texto, el Hypertexto puede estar formado por tablas, imágenes u otros elementos. En esencia, HTML sirve para estructurar documentos (títulos, párrafos, listas, etc.), pero no escribe la apariencia o el diseño de un documento sino que ofrece las herramientas necesarias para dar un formato, según la capacidad al servidor web en el que se almacenan la página web y la capacidad del navegador (Tamaño de la pantalla, fuentes que tiene instaladas, etc. (Vértice, 2015)

```

<!DOCTYPE html>
<html>
  <head>
    <title>YouTube HTML5 Demo</title>
    <link rel="stylesheet" href="/demo/demo.css" type="text/css">
    <script type="text/javascript" src="/demo/demo.js"></script>
  </head>
  <body>
    <header id="global-header">
      <h1 id="global-title">HTML5 Demo</h1>
      <a href="/"></a>
    </header>
    <nav id="global-nav">
      <a href="/" class="nav-button">Home</a>
      <a href="/my_subscriptions?masthead=1" class="nav-button">Subscriptions</a>
      <a href="/browse" class="nav-button">Videos</a>
      <a href="/shows" class="nav-button">Shows</a>
      <a href="/members" class="nav-button">Channels</a>
      <form id="search-form">
        <input id="search-textbox" type="search" name="query">
        <button id="search-button" type="submit" class="blue-button">Search</button>
      </form>
      <button id="upload-button" class="yellow-button" type="button">Upload</button>
    </nav>
    <article id="watch-page">
      <h1>O3D Beach Demo</h1>
      <section id="main-content">
        <div id="video-player">
          <video width="640" height="360" src="/demo/google_main.mp4?2" autobuffer>
            <div class="video-fallback">
              <br>You must have an HTML5 capable browser.
            </div>
          </video>
          <div class="controls">

```

Figura 2. Ejemplo de código HTML

Fuente: Equipo de Investigación

9.4.3. Cascading Style Sheets CSS

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille. (Ojeda, 2015)

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando así la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos. (Silva & Mercerat, 2011)

Puede decirse que el lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre

corchetes y consiste en una propiedad o atributo, y un valor separados por dos puntos. (Christopher, 2012)

Se trata de una sintaxis sencilla, estandarizada, que proporciona a los diseñadores un amplio control sobre la presentación de sus páginas Web. De este modo, CSS representa ya un componente esencial del diseño Web en la actualidad. (Schmitt, 2013)

Las CSS constituyen un mecanismo para asociar estilos de composición a documentos estructurados, del tipo HTML o XML. Aplicables a cualquier navegador, admiten un mayor control sobre los distintos elementos de una página, permitiendo definir el estilo de las fuentes, el color, el espaciado del texto, la posición del contenido, e incluso variaciones en el sonido en los elementos auditivos. Estos estilos pueden definirse para luego ser aplicados al código de cualquier documento. (Briggs, Champeon, Costello, & Patterson, 2013)

```

64 #sublinks_foro {
65     margin: 2px 0 0 -40px;
66     padding: 5px;
67     width: 750px;
68     border: 5px solid #549DC9;
69 }
70
71 #sumolari_proyectos, #sublinks_foro, #main_content .post-stick-extra-info {
72     border-radius: 10px;
73     -webkit-border-radius: 10px;
74     -moz-border-radius: 10px;
75     background: #FFF;
76     position: absolute;
77     z-index: 999999;
78     display: none;
79 }
80
81 #main_content .post-stick-extra-info {
82     top: -110px;
83     padding: 5px;
84     width: 400px;
85     font-family: "Droid Sans", Verdana, Geneva, sans-serif;
86     font-size: 0.85em;
87     line-height: 1.5em;
88     border: 5px solid #FFF;
89     background: #549DC9;
90 }
91
92 #main_content p.post-stick-extra-info-title {
93     position: relative;
94     left: -10px;
95     padding: 0 10px 10px 10px;
96     width: 390px;

```

Figura 3. Ejemplo de código CSS

Fuente: Equipo de Investigación

Los diseñadores han encontrado una herramienta con ventajas para mejorar su trabajo, donde destacan la posibilidad de acceder a las propiedades de los objetos de manera dinámica, y el hecho de poder modificar elementos a los que antes no se tenía acceso,

como son las barras de desplazamiento del navegador, los elementos de formulario o el mismo puntero del ratón (Briggs, Champeon, Costello, & Patterson, 2013)

9.5. JavaScript

JavaScript es el lenguaje interpretado más utilizado, principalmente en la construcción de páginas Web, con una sintaxis muy semejante a Java y a C. Pero, al contrario que Java, no se trata de un lenguaje orientado a objetos propiamente dicho, sino que éste está basado en prototipos, ya que las nuevas clases se generan mediante la clonación de las clases base (prototipos) y extendiendo su funcionalidad. (Flanagan, 2015)

9.5.1. Especificaciones oficiales

ECMA ha publicado varios estándares relacionados con ECMAScript. En Junio de 1997 se publicó la primera edición del estándar ECMA-262. Un año después, en Junio de 1998 se realizaron pequeñas modificaciones para adaptarlo al estándar ISO/IEC-16262 y se creó la segunda edición. La tercera edición del estándar ECMA-262 (publicada en Diciembre de 1999) es la versión que utilizan los navegadores. Actualmente se encuentra en desarrollo la cuarta versión de ECMA-262, que podría incluir novedades como paquetes, namespaces, definición explícita de clases, etc. ECMA también ha definido varios estándares relacionados con ECMAScript, como el estándar ECMA-357, que define una extensión conocida como E4X y que permite la integración de JavaScript y XML. (Stothard, 2011)

El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento. Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript dentro de la cabecera del documento (dentro de la etiqueta `<head>`), un ejemplo de esto se muestra en la Figura 2.

Además de los formularios y de todas las funciones y utilidades de JavaScript que se han visto, existen muchas otras utilidades que es necesario conocer para desarrollar aplicaciones completas. Como no es posible estudiar todas las herramientas que se pueden crear con JavaScript, a continuación se muestran algunas de las utilidades básicas más comunes. (Gauchat, 2012)

```

var Person = function (firstName, lastName, age) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
};

person = new Person("John", "Doe", "50");

Person.prototype.sayHello = function () {
    alert("Hello " + this.firstName + " " + this.lastName);
};

person.sayHello();

```

Figura 4. Ejemplo de código JavaScript

Fuente: Equipo de Investigación

JavaScript, al igual que Flash, Visual Basic Script, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML (lenguaje para el diseño de páginas de Internet). Al ser la más sencilla, es por el momento la más extendida. JavaScript no es un lenguaje de programación propiamente dicho como C, C++, Delphi, etc. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y planillas de cálculo. No se puede desarrollar un programa con JavaScript que se ejecute fuera de un Navegador, aunque en este momento comienza a expandirse a otras áreas como la programación en el servidor con Node.js (Crockford, 2016)

9.6. PHP: Hypertext Preprocessor

9.6.1. Definiciones fundamentales de PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP". (Castagnetto, 2015)

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. (Puertas, 2015)

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts. (Cobo, 2015)

A menudo los desarrolladores, especialmente aquellos que trabajan principalmente con PHP o Java, discuten sobre la superioridad de un lenguaje u otro. Estas discusiones pueden llegar a ser algo tensas, y los comentaristas escriben a un alto nivel técnico. Así que usted podría pensar, y no le culparíamos por ello, que estos lenguajes son estructuralmente iguales y mutuamente excluyentes. (Diógenes, 2016)

```

1  <?php
2  require('fpdf.php');
3  require_once('barcode.inc.php');
4  require_once("conecta.php");
5
6  $numero=$_GET["f"];
7
8  $consul="select categoria, descripcion from productos where codigo=$numero";
9  $r=mysql_query($consul);
10 $dato=mysql_fetch_row($r);
11 $llenar="00000".$numero;
12 new barCodeGenerator($llenar,1,'test.gif',190, 130, VERDADERO);
13
14
15 $pdf = new FPDF('P','cm',array(4,5));
16 $pdf->AddPage();
17     $pdf->SetFont('Arial','',7);
18     $informacion="$dato[0] $dato[1]";
19     $pdf->Cell(3.9,1,$informacion,1,0,'L');
20
21     $pdf->Image('test.gif',1,2,3,1);
22
23
24 $pdf->Output();
25 ?>
26

```

Figura 5. Ejemplo de código PHP

Fuente: Equipo de Investigación

Una de las principales diferencias estructurales entre PHP y Java es la diferencia entre un lenguaje de tipado fuerte y uno de tipado débil. Java es un lenguaje de tipado fuerte, lo que significa que requiere declaraciones explícitas para funcionar y esto es

respaldado por el compilador. A un mayor nivel, se puede decir que tiene estrictas expectativas de cómo se expresan las entradas y las salidas. Si esas expectativas no se cumplen, el compilador fallará y el programa no funcionará hasta que se resuelvan los errores. (Silva & Mercerat, 2011)

9.6.2. Código PHP en una página WEB HTML

Un archivo PHP normalmente suele contener HTML y trozos de código PHP. Por ello antes de estudiar PHP es necesario haber estudiado HTML. A veces, se dice que PHP es un lenguaje “embebido” en HTML, es decir, incrustado en código HTML. Todos los fragmentos de código PHP deben comenzar con la siguiente cadena de caracteres: `<?php` , y finalizar con `?>`.

```
<HTML>
<HEAD>
  <TITLE>Trabajando con Cadenas</TITLE>
</HEAD>
<BODY>
  <CENTER>
    <H2>Funciones <I>strnatcmp</I></H2>
    <?php
      $cad1="10000 Pts";
      $cad2="2500 Pts";
      echo "<TABLE BORDER='1' CELLPADDING='2' CELLSPACING='2'>\n";
      echo "<TR ALIGN='center'><TD BGCOLOR='yellow'>cadena1</TD>";
      echo "<TD>$cad1</TD></TR>\n";
      echo "<TR ALIGN='center'><TD BGCOLOR='yellow'>cadena2</TD>";
      echo "<TD>$cad2</TD></TR>\n";
      echo "<TR ALIGN='center'>";
      echo "<TD BGCOLOR='yellow'>strcmp(cadena1,cadena2)</TD>";
      echo "<TD>".strcmp($cad1,$cad2)."</TD></TR>\n";
      echo "<TR ALIGN='center'>";
      echo "<TD BGCOLOR='yellow'>strnatcmp(cadena1,cadena2)</TD>";
      echo "<TD>".strnatcmp($cad1,$cad2)."</TD></TR>\n";
      echo "</TABLE>\n";
    ?>
  </CENTER>
</BODY>
</HTML>
```

Figura 6. Código PHP insertado en el código HTML

Fuente: Equipo de Investigación

El complemento de multiplexación de conexiones de mysqlnd está implementado como una extensión de PHP. Está escrito en C y opera bajo el manto de PHP. Durante el arranque del intérprete de PHP, en la fase de inicialización de módulos del motor de PHP, se registra como un complemento de mysqlnd para reemplazar a los métodos en C específicos de mysqlnd. (Welling & Thomson, 2014)

9.6.3. ASP.NET

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.(Msdn.microsoft.com, 2017)

Este lenguaje incluye:

- Marco de trabajo de página y controles
- Compilador de ASP.NET
- Infraestructura de seguridad
- Funciones de administración de estado
- Configuración de la aplicación
- Supervisión de estado y características de rendimiento
- Capacidad de depuración
- Marco de trabajo de servicios Web XML
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones
- Entorno de diseñador extensible

9.6.4. Marco de trabajo de ASP.Net

El marco de trabajo de páginas y controles ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor Web para generar y representar de forma dinámica páginas Web ASP.NET. Las páginas Web ASP.NET se pueden solicitar a cualquier explorador o dispositivo del cliente y ASP.NET representa el marcado (como HTML) al explorador que realizó la solicitud. Como norma, puede utilizar la misma página para varios exploradores, porque ASP.NET representa el marcado adecuado para el explorador que realiza la solicitud. Sin embargo, puede diseñar una página Web ASP.NET para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 6, y aprovechar así todas las características de ese explorador. ASP.NET es

compatible con los controles móviles de los dispositivos preparados para trabajar en Web como teléfonos celulares, PC portátiles y asistentes digitales personales (PDA). (Prosky, Nils-Georg, Furda, & DeVries, 2004)

Las páginas Web ASP.NET están completamente orientadas a objetos. En las páginas Web ASP.NET se puede trabajar con elementos HTML que usen propiedades, métodos y eventos. El marco de trabajo de páginas ASP.NET quita los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página. Para obtener más información, vea Información general sobre páginas Web ASP.NET. (Msdn.microsoft.com, 2017)

```
Protected Sub grdUsuarios_RowDataBound(ByVal sender As Object, ByVal e As System.Web.UI.Web
    Select Case e.Row.RowType
        Case DataControlRowType.DataRow
            Dim intStatus As Integer = CType(e.Row.Cells(2).Text, Integer)
            Dim strImageUrl As String
            Select Case intStatus
                Case 1
                    strImageUrl = "img/ico_vermelho.gif"
                Case 2
                    strImageUrl = "img/ico_verde.gif"
                Case Else
                    strImageUrl = "img/ico_azul.gif"
            End Select

            Dim imgIcôneStatus As WebControls.Image = e.Row.FindControl("imgIcôneStatus")
            imgIcôneStatus.ImageUrl = strImageUrl

        End Select
    End Select
```

Figura 7. Código PHP insertado en el código HTML

Fuente: Equipo de Investigación

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Los controles se escriben una vez, se pueden utilizar en varias páginas y se integran en la página Web ASP.NET en la que se colocan durante la representación. (Msdn.microsoft.com, 2017)

9.7. Sistemas de gestión de Bases de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a los mismos. Esta definición es prácticamente idéntica a la que se dio anteriormente de Sistema de Información, de hecho normalmente en el núcleo de un SI se sitúa un SGBD. El caso de lo SIG es un poco diferente ya que en principio las bases de datos espaciales no son adecuadas para su manejo con SGBD tradicionales. (Connolly & Begg, 2015)

Sin embargo, a lo largo del desarrollo de las tecnologías ligadas a los SIG desde los setenta hasta la actualidad, una de las tendencias más claras es el papel, cada vez más importante, que tiene el uso de SGBD para la gestión de datos temáticos como apoyo al SIG. En principio se utilizaron para almacenar los atributos temáticos asociados a un conjunto de entidades espaciales almacenadas en formato vectorial, hoy en día se están empezando a utilizar además para el almacenamiento de la información geométrica (conjunto de coordenadas) de las entidades espaciales. Aunque se han hecho algunos intentos para almacenar información en formato raster en un SGBD, esta opción no resulta eficiente. (Elmasri, Navathe, & Castillo, 2012)

Un SGBD permite el almacenamiento, manipulación y consulta de datos pertenecientes a una base de datos organizada en uno o varios ficheros. En el modelo más extendido (base de datos relacional, que puede verse en Figura 8) la base de datos consiste, de cara al usuario, en un conjunto de tablas entre las que se establecen relaciones. (Date, Faudón, & Ruiz, 2011)

Es el modelo más utilizado hoy en día. Una base de datos relacional es básicamente un conjunto de tablas, similares a las tablas de una hoja de cálculo, formadas por filas (registros) y columnas (campos). Los registros representan cada uno de los objetos descritos en la tabla y los campos los atributos (variables de cualquier tipo) de los objetos. En el modelo relacional de base de datos, las tablas comparten algún campo entre ellas. Estos campos compartidos van a servir para establecer relaciones entre las tablas que permitan consultas complejas. (Korth & Silberschatz, 2013)

9.7.1. MySQL

MySQL (My Structured Query Language) MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos. Actualmente es la base de datos de código abierto más popular del mundo. Con su rendimiento, confiabilidad y facilidad de uso comprobados, MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la Web, utilizada por propiedades web de alto perfil como Facebook, Twitter, YouTube, y los cinco principales sitios web*. Además, es una alternativa extremadamente popular como base de datos integrada, distribuida por miles de ISV y OEM. (Welling & Thomson, 2014)

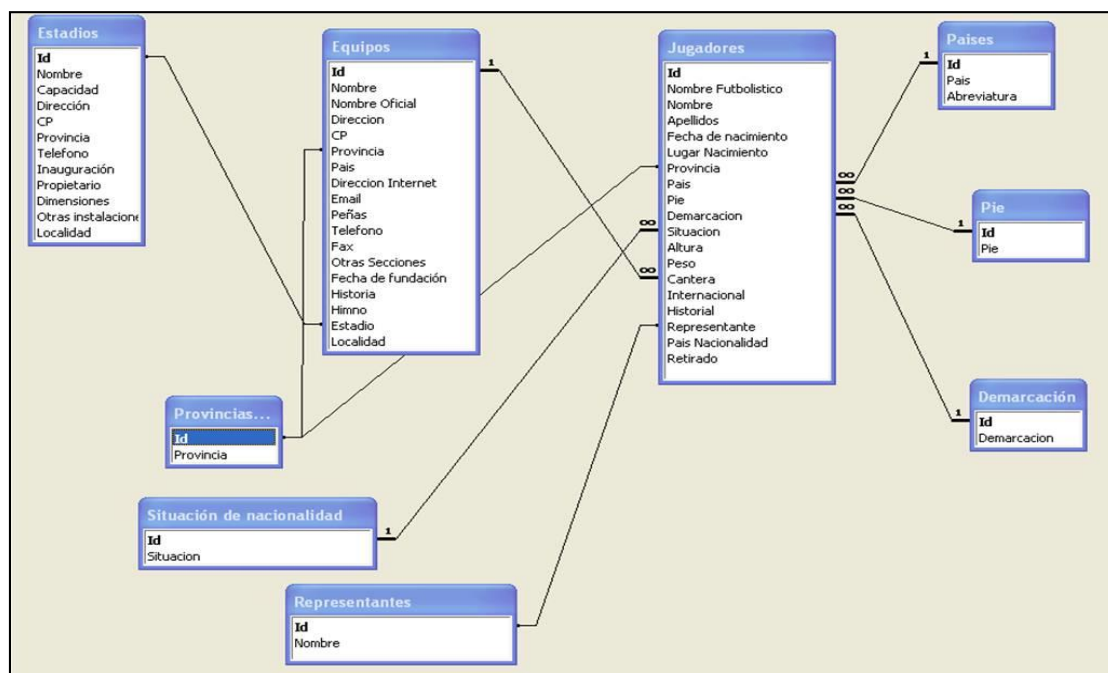


Figura 8. Ejemplo del Modelo relacional

Fuente: Equipo de Investigación

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet. (Pérez López, 2016)

MySQL presenta las siguientes ventajas:

1. MySQL software es Open Source
2. Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
3. Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
4. Facilidad de configuración e instalación.
5. Soporta gran variedad de Sistemas Operativos
6. Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
7. Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet
8. El software MySQL usa la licencia GPL

Y además, presenta las siguientes desventajas:

1. Un gran porcentaje de las utilidades de MySQL no están documentadas.
2. No es intuitivo, como otros programas (MS ACCESS).

La arquitectura de MySQL tiene como característica más notable el separar el motor de almacenamiento (que se encarga de los detalles de entrada-salida y representación de la información en memoria secundaria) del resto de los componentes de la arquitectura. Es decir, el diseño del gestor está preparado para que se pueda cambiar el gestor de almacenamiento. Esto permite incluso crear nuevos motores de almacenamiento especializados para ciertas tareas o tipos de aplicaciones. (Zawodny & Balling, 2014)

La Figura 9 es una visión abstracta de la arquitectura lógica de MySQL. La figura hace una división entre los componentes que conforman el servidor, las aplicaciones cliente que lo utilizan y las partes del sistema operativo en las que se basa el almacenamiento físico.

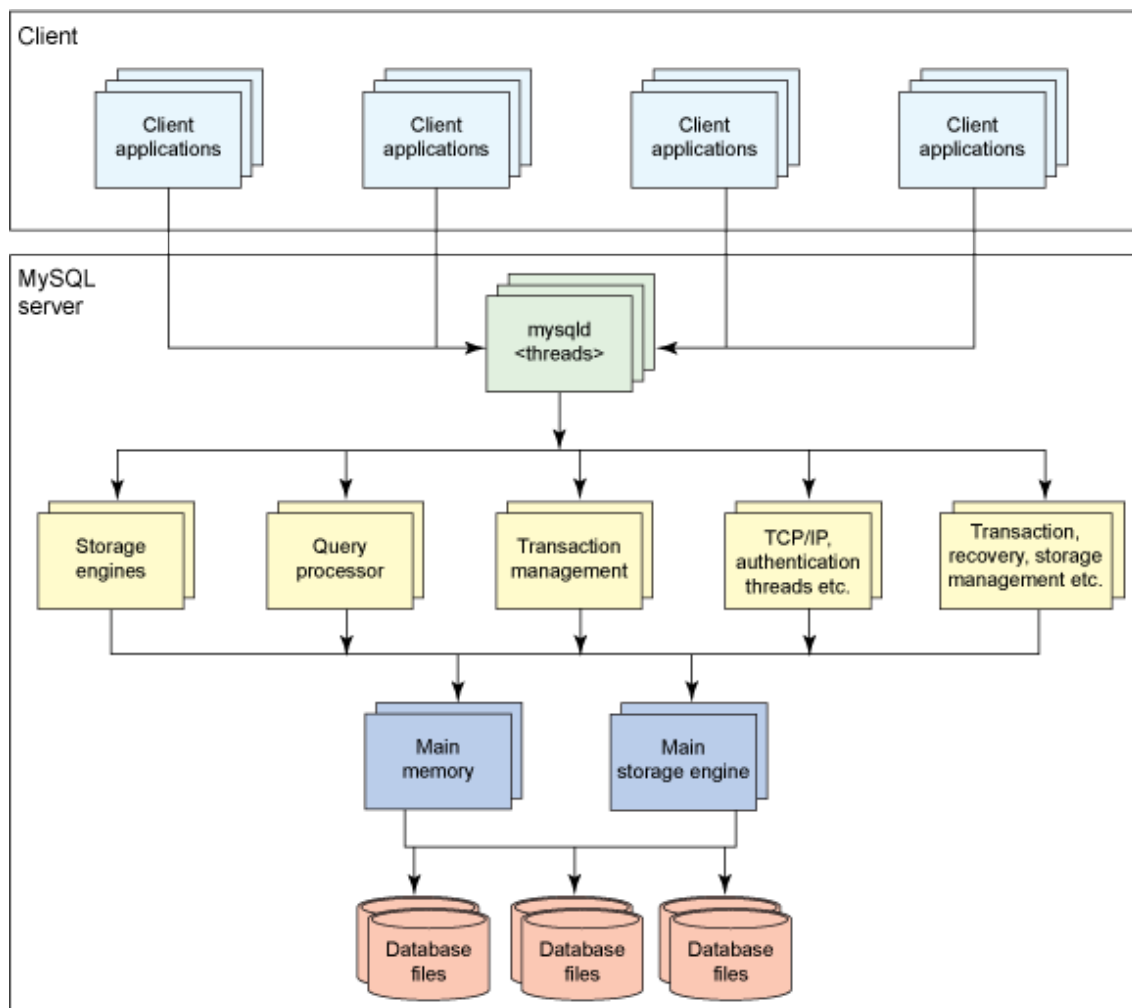


Figura 9. Arquitectura funcional de MySQL

Fuente: Equipo de Investigación

9.7.2. Microsoft SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact -SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas. (Agrawal, Chaudhuri, & Kollar, 2005)

Bajo el nombre código Yukon en su etapa de desarrollo, SQL Server 2005 fue lanzado en noviembre de 2005. Se dice que el producto 2005 proporcionó una mayor flexibilidad, escalabilidad, confiabilidad y seguridad a las aplicaciones de base de datos, y permitió que fueran más fáciles de crear y desplegar, lo que reduce la complejidad y el

tedio involucrado en la gestión de bases de datos. SQL Server 2005 también incluía más soporte administrativo.(Cobo, 2015)

El código original de SQL Server ha sido desarrollado por Sybase; a finales de 1980, Microsoft, Sybase y Ashton-Tate colaboraron para producir la primera versión del producto, SQL Server 4.2 para OS/2. Posteriormente, tanto Sybase como Microsoft ofrecieron productos de SQL Server. Sybase cambió después el nombre de su producto a Adaptive Server Enterprise.(Puertas, 2015)

Hoy día Microsoft SQL Server es una plataforma de base de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico; es también una plataforma de Business Intelligence para soluciones de integración, análisis y creación de informes de datos.

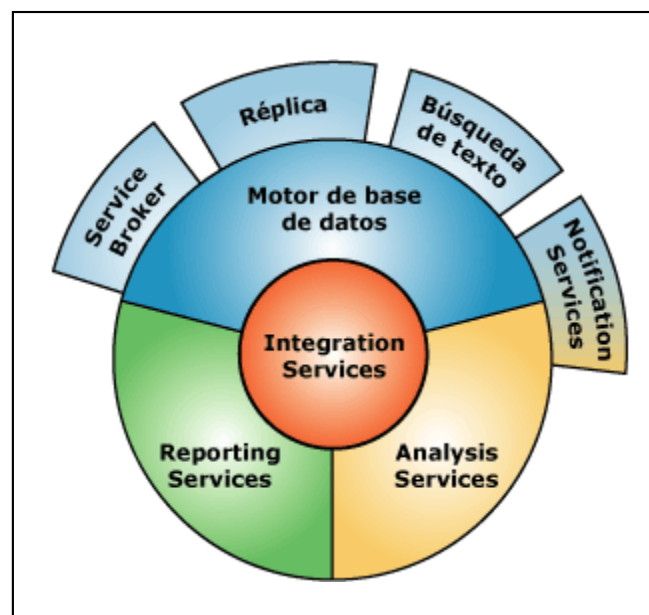


Figura 10. Arquitectura funcional de MySQL

Fuente: Equipo de Investigación

SQL Server 2005 introduce "estudios" que le ayudarán en las tareas de programación y administración: SQL Server Management Studio y Business Intelligence Development Studio. En Management Studio, se desarrolla y administra SQL Server Database Engine (Motor de base de datos de SQL Server) y soluciones de notificación, se administran las soluciones de Analysis Services implementadas, se administran y ejecutan los paquetes de Integration Services, y se administran los servidores de informes y los informes y modelos de informe de Reporting Services. En BI Development Studio, se desarrollan

soluciones de Business Intelligence mediante proyectos de Analysis Services para desarrollar cubos, dimensiones y estructuras de minería; se crean proyectos de Reporting Services para crear informes; se crea el modelo de informes para definir modelos para los informes y se desarrollan proyectos de Integration Services para crear paquetes.(Buffington, 2010)

9.7.3. Ventajas de Microsoft SQL Server

1. Es un sistema de gestión de base de datos.
2. Es útil para manejar y obtener datos de la red de redes.
3. Nos permite olvidarnos de los ficheros que forman la base de datos.
4. Si trabajamos en una red social nos permite agregar otros servidores de SQL Server. Por ejemplo dos personas que trabajan con SQL Server, uno de ellos se puede conectar al servidor de su otro compañero y así se puede ver las bases de datos del otro compañero con SQL Server.
5. SQL permite administrar permisos a todo. También permite que alguien conecte su SQLO al nuestro pero sin embargo podemos decirle que no puede ver esta base de datos pero otro sí.

Una vez analizados en detalles los requerimientos funcionales del sistema informático objeto de esta investigación y las características, ventajas y desventajas de los principales SGBD de los que se disponen en la actualidad, se determina, que para el funcionamiento óptimo del sistema, se utilice Microsoft SQL Server como motor de base de datos. (Agrawal, Chaudhuri, & Kollar, 2005)

9.8. Frameworks para el desarrollo WEB

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírsenos. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica

incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (González & Romero, 2012)

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (Catalani, 2014)

Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el **patrón MVC**. El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (Gutiérrez, 2014)

9.8.1. **Symfony 2**

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (Eguiluz, 2013)

Symfony está desarrollado completamente con PHP y ha sido probado con éxito en sitios como Yahoo! Answers, delicious, DailyMotion y muchos otros sitios web de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características. (Cecilia Rodríguez, 2015)

Características de Symfony:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows, Mac y Unix estándares)
- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

Symfony considera un proyecto como "un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio y que comparten el mismo modelo de objetos". Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones. Normalmente, una aplicación se ejecuta de forma independiente respecto de otras aplicaciones del mismo proyecto. Lo habitual es que un proyecto contenga dos aplicaciones: una para la parte pública y otra para la parte de gestión, compartiendo ambas la misma base de datos. También es posible definir proyectos que estén formados por varios sitios web pequeños, cada uno de ellos considerado como una aplicación. En este caso, es importante tener en cuenta que los enlaces entre aplicaciones se deben indicar de forma absoluta. (Vilas Usó, 2015)

Cada aplicación está formada por uno o más módulos. Un módulo normalmente representa a una página web o a un grupo de páginas con un propósito relacionado. Por ejemplo, una aplicación podría tener módulos como home, artículos, ayuda, carrito de compra, cuenta, etc. Los módulos almacenan las acciones, que representan cada una de las operaciones que se puede realizar en un módulo. Por ejemplo el módulo carrito de compra puede definir acciones como añadir, mostrar y actualizar. Normalmente las

acciones se describen mediante verbos. Trabajar con acciones es muy similar a trabajar con las páginas de una aplicación web tradicional, aunque en este caso dos acciones diferentes pueden acabar mostrando la misma página (como por ejemplo la acción de añadir un comentario a una entrada de un blog, que acaba volviendo a mostrar la página de la entrada con el nuevo comentario). (Zubillaga Beraza, 2016)

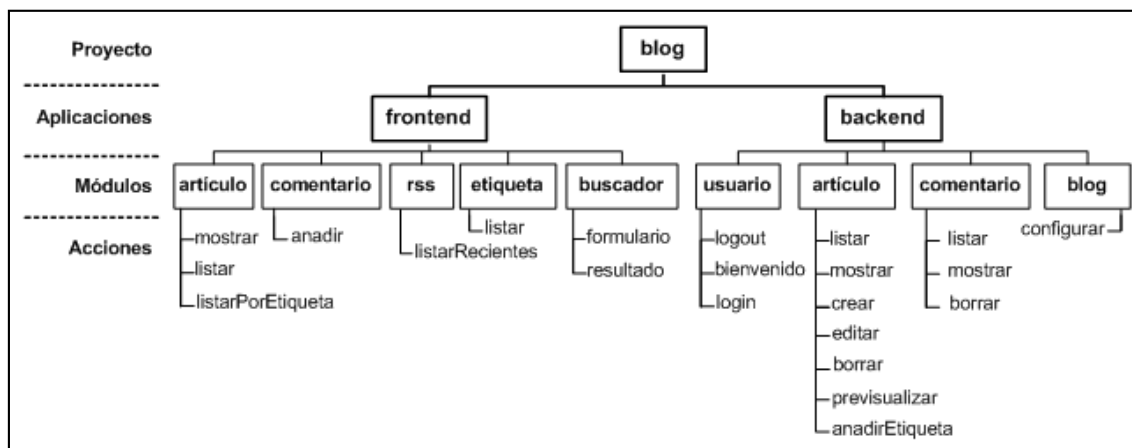


Figura 11. Organización del código para un proyecto de un blog en Symfony

Fuente: Equipo de Investigación

La Figura 11 muestra un ejemplo de organización del código para un proyecto de un blog, siguiendo la estructura de proyecto / aplicación / módulo / acción. No obstante, la estructura de directorios real del proyecto es diferente al esquema mostrado por esa figura.

9.8.2. NetBeans

NetBeans IDE es un entorno de desarrollo integrado (IDE), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. (Boudreau, Greene, Spurlin, & Woehr, 2012)

Netbeans (Figura 12) es un entorno de desarrollo gratuito y de código abierto que en el momento de escribir este artículo está en su versión 7.4. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a las siguientes tecnologías, entre otras: Java, PHP,

Groovy, C/C++, HTML5,... Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS, etc. (Böck, 2011)

Características principales

Suele dar soporte a casi todas las novedades en el lenguaje Java. Cualquier preview del lenguaje es rápidamente soportada por Netbeans. Asistentes para la creación y configuración de distintos proyectos, incluida la elección de algunos frameworks. Buen editor de código, multilenguaje, con el habitual coloreado y sugerencias de código, acceso a clases haciendo clic en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, plantillas de código, etc. También hay tecnologías donde podemos usar el pulsar y arrastrar para incluir componentes en nuestro código. (Böck, 2011)

Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario. Una vez que nos accedemos al código de una clase java, por poner un ejemplo, se nos mostrarán distintas ventanas con dicho código, su localización en el proyecto, una lista de los métodos y propiedades (ordenadas alfabéticamente), también hay una vista que nos presenta las jerarquías que tiene nuestra clase y otras muchas opciones. Por supuesto personalizable según el gusto de cada usuario. (Boudreau, Greene, Spurlin, & Woehr, 2012)

Incluye herramientas para depurado de errores: el debugger que incluye el IDE es bastante útil para encontrar dónde fallan las cosas. Podemos definir puntos de ruptura en la línea de código que nos interese, monitorizar en tiempo real los valores de propiedades y variables, se nos permite ir paso a paso, ejecutar un método de un tirón, o entrar dentro, en fin, las opciones típicas, pero que tan útiles son en el trabajo diario. Incluso podemos usar el debugger en tiempo de ejecución, conectándonos a él cuándo ya tenemos un proceso ejecutándose. (Hartati, 2016)

Optimización de código: por su parte el Profiler nos ayuda a optimizar nuestras aplicaciones e intentar hacer que se ejecuten más rápido y con el mínimo uso de memoria. Podemos igualmente configurarlo a nuestro gusto, aunque por defecto, nos ofrece opciones bastante útiles. Lo importante es que podemos ver el comportamiento de nuestra aplicación y obtener indicadores e información de cómo y cuantos recursos

consume, cuantos objetos se crean, también podemos obtener capturas del estado del sistema en diferentes momentos (Snapshots) y compararlos entre sí. (Petri, 2011)

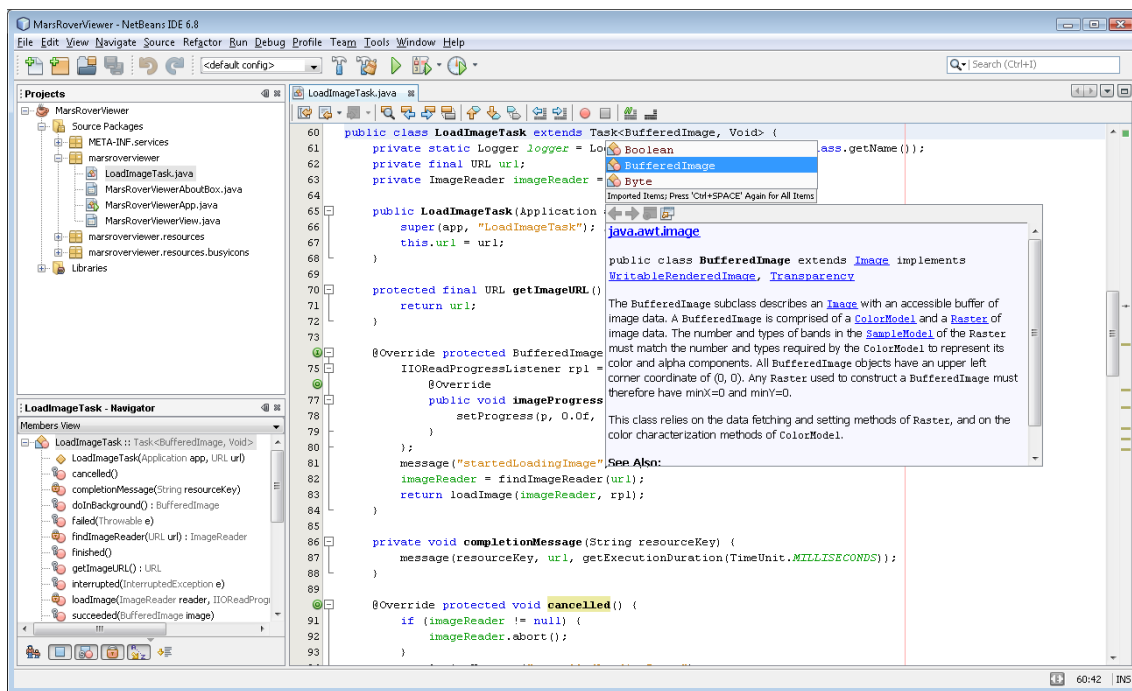


Figura 12. Interfaz principal de NetBeans

Fuente: Equipo de Investigación

Acceso a base de datos: desde el propio NetBeans podemos conectarnos a distintos sistemas gestores de bases de datos, como pueden ser Oracle, MySQL y demás, y ver las tablas, realizar consultas y modificaciones, y todo ello integrado en el propio IDE. Se integra con diversos servidores de aplicaciones, de tal manera que podemos gestionarlos desde el propio IDE: inicio, parada, arranque en modo debug, despliegues. Entre otros podemos usar Apache Tomcat, GlassFish, JBoss, WebLogic, Sailfin, Sun Java System Application Server, etc. (Böck, 2011)

9.8.3. Microsoft Visual Studio

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes

utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.

Visual Studio proporciona compatibilidad con el enlace en tiempo de ejecución a los tipos dinámicos ya que introduce un nuevo tipo, `dynamic`. Este nuevo tipo permite muchos escenarios nuevos, incluido el acceso simplificado a API de COM tales como las API de automatización de Office, a API dinámicas como las bibliotecas de IronPython, y a DOM (Document Object Model) HTML. Para obtener más información, vea *Uso de tipo dinámico (Guía de programación de C#)* y *dynamic (Referencia de C#)*.

Un aspecto a tener en cuenta es que Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros. Este sistema está disponible en el mercado desde versión 2005 de este mismo año, Microsoft ofrece gratuitamente las Ediciones Express, que son versiones básicas separadas por lenguajes de programación o plataforma enfocadas a estudiantes y programación amateur. Estas ediciones son iguales que el entorno de desarrollo comercial, pero sin características avanzadas de integración. Dichas ediciones son:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual Web Developer Express Edition (para programar en ASP.NET)
- Visual F# (Apareció en Visual Studio 2010, es parecido al J#)*
- Windows Phone 8 SDK
- Windows Azure SDK

El acceso a las interfaces COM, incluidas las API de automatización de Office, se ha mejorado considerablemente ya que se han agregado argumentos con nombre y opcionales, el tipo `dynamic`, propiedades indizadas y modificadores `ref` opcionales.

Ahora se puede implementar una aplicación que contiene información de tipos incrustada en lugar de información de tipos que se importa desde un ensamblado de interoperabilidad primario (PIA). Con la información de tipos incrustada, su aplicación puede usar tipos de un ensamblado sin necesidad de establecer referencias al ensamblado del runtime o al ensamblado de interoperabilidad primario (PIA). Si se publican varias versiones del ensamblado del runtime, la aplicación que contiene la información de tipos incrustada puede funcionar con las diferentes versiones sin que sea necesario volver a compilarla.

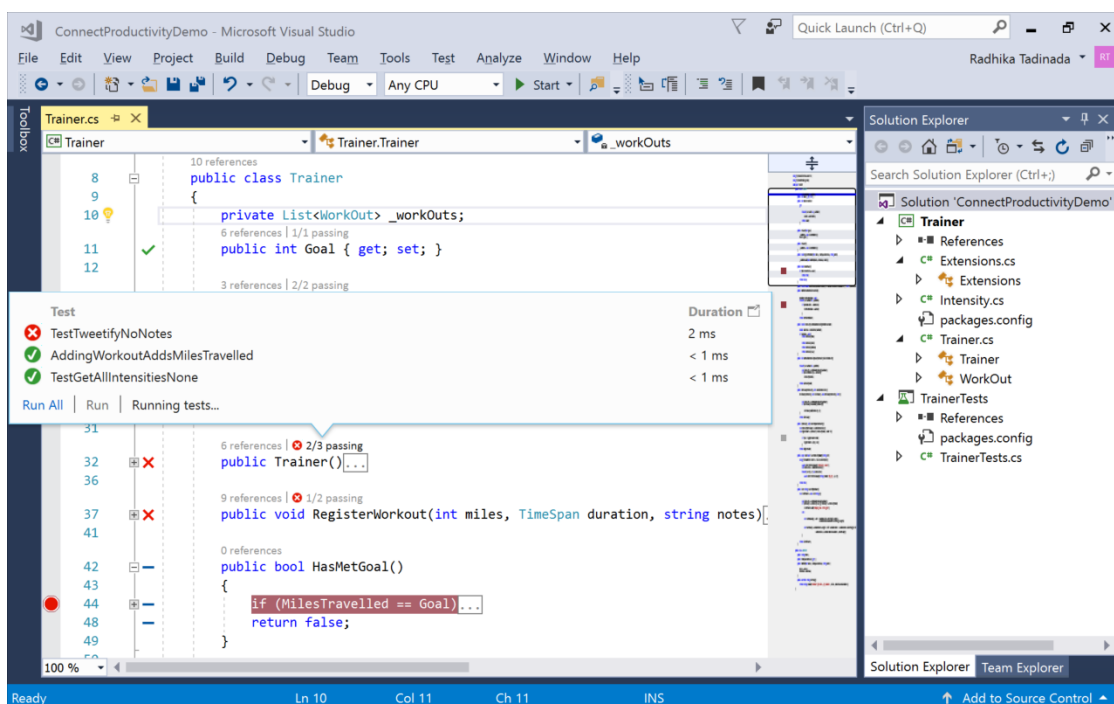


Figura 13. Interfaz principal del software Visual Studio

Fuente: Equipo de Investigación

La covarianza permite usar un tipo más derivado que el especificado por el parámetro genérico, mientras que la contravarianza permite utilizar un tipo menos derivado. Esto permite la conversión implícita de las clases que implementan interfaces variantes y proporciona mayor flexibilidad a la hora de hacer coincidir las firmas de método con tipos de delegado variantes. Las nuevas palabras clave `in` y `out` del lenguaje permiten crear interfaces y delegados variantes. .NET Framework también incluye compatibilidad con la varianza para varios delegados e interfaces genéricos existentes, incluidos la interfaz `IEnumerable<T>` y los delegados `Action<T>` y `Func<TResult>`.

9.8.4. Resultado del proceso de selección de las herramientas para el desarrollo

Luego de analizada toda la documentación sobre las principales herramientas de las que se dispone en la actualidad para complementar el desarrollo del software objeto de la presente investigación, además de comprobado su funcionamiento y rendimiento el equipos con iguales características de software y hardware, se determina que **Microsoft Visual Studio 2013** se adecua mejor a los requerimientos para la implementación del sistema, además se determina que la utilización de **Microsoft SQL Server** como gestor de la bases de dato del sistema permitirá una alta eficiencia y rapidez en el desarrollo.

9.9. Herramientas auxiliares para de desarrollo WEB

El diseño efectivo y eficiente de sitios web puede ser una tarea compleja, difícil y larga. De ahí que las herramientas se conviertan en un elemento clave a la hora de maximizar nuestros recursos. Hoy en día el mercado de la herramientas es muy amplio y cubre todo tipo de necesidades, por eso resulta difícil escoger la adecuada para cada proyecto.

Seguidamente se presentarán, algunas de las herramientas empleadas en el desarrollo del sistema WEB resultado de esta investigación.

9.9.1. Sublime Text

Sublime Text es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra nuestra atención completamente. Sublime Text permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla. (Skinner, 2015)

El programa cuenta “de serie” con 22 combinaciones de color posibles, aunque se pueden conseguir más. Para navegar por el código cuenta con Minimap, un panel que permite moverse por el código de forma rápida como puede verse en la Figura 14.

El sistema de resaltado de sintaxis de Sublime Text soporta un gran número de lenguajes (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby,

SQL, TCL, Textile and XML). El programa dispone de auto-guardado, muchas opciones de personalización, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, Snippets y auto completar, entre otras funcionalidades. Algunas de sus características son ampliables mediante plugins. (Skinner, 2015)

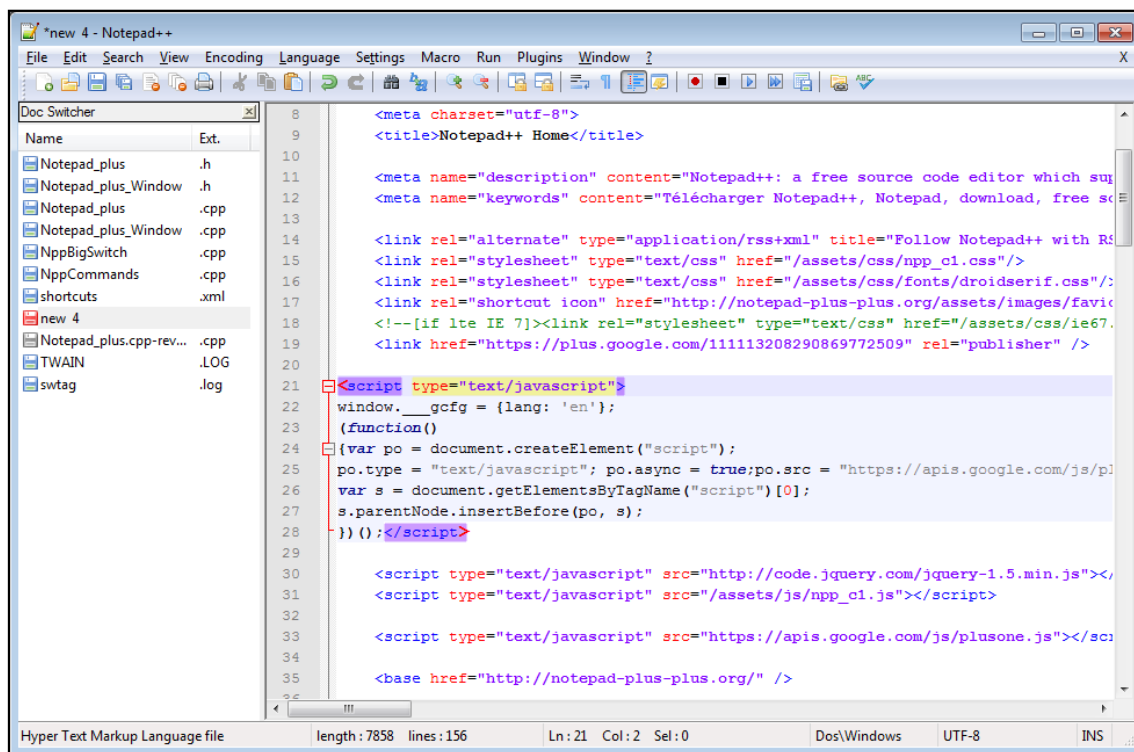


Figura 14. Notepad++

Fuente: Equipo de Investigación

Sublime Text es un programa de pago, aunque se puede descargar una versión de prueba, plenamente funcional y sin limitación de tiempo. La licencia individual cuesta 59 dólares. Cada programador es un pequeño maniático con sus credos y sus fobias respecto de las herramientas que emplea, pero si lo que quieres es centrarte únicamente en el código, tal vez deberías probar Sublime Text. La aplicación está disponible para OS X, Linux y Windows. (Rodríguez, 2012)

9.9.2. WampServer

WampServer es un entorno de desarrollo web para Windows en el cual se podrán crear aplicaciones web con Apache, PHP y base de datos en MySQL (motor de base de datos). Esta herramienta incluye además con un administrador de base de datos

PHPMyAdmin con el cual podremos crear una nueva base de datos e ingresar la data de las tablas creadas en ella, realizar consultas y generar scripts SQL, como exportar e importar scripts de base de datos. WampServer ofrece a los desarrolladores herramientas necesarias para realizar aplicaciones web de manera local, con un sistema operativo (Windows), un manejador de base de datos (MySQL), un software de programación script web PHP. WampSever se caracteriza por que puede ser usado de forma libre es decir no debemos de contar con alguna licencia el cual nos permita el uso de la misma, ya que pertenece a la corriente de "open source". (Agrawal & Gupta, 2014)

Utilidad

Su utilidad es importante a la hora de desarrollar aplicaciones web, ya que funciona al igual como si cuando trabajamos en un servidor web, ya que podemos ejecutar estas aplicaciones de manera local y ver como sería el funcionamiento antes de ser subidas a un hosting o servidor web. Además de ello podemos gestionar datos con la ayuda del motor de base de datos (MySQL) y su administrador (PHPMyAdmin). (Agrawal & Gupta, 2014)

Características

- Pre visualizar sitios web locamente
- Instalar un servidor web en Windows
- Administrar configuraciones de servidores Apache
- Crear aplicaciones web
- Gestionar bases de datos MySQL
- Utilizar lenguaje PHP
- Ejecutar archivos .PHP localmente
- Realizar pruebas con sitios PHP antes de subirlos a Internet

Ventajas:

- Interfaz intuitiva y permite poder trabajar con versiones anteriores.
- La instalación modificará los archivos de configuración (*.conf) con la ruta donde finalmente se ubicará el programa. También crea un directorio denominado 'WWW' que será la raíz para tus documentos

Desventajas:

- Requiere conocimientos previos de Apache, PHP y MySQL.
- Todos los componentes vienen instalados y cualquier usuario puede ingresar y alterar los códigos.

10. Hipótesis

Se optimizará el proceso de evaluación de desempeño docente en la Universidad Técnica de Cotopaxi mediante la implementación de un sistema informático de gestión de información.

11. Metodología**11.1. Tipos de investigación****11.1.1. Investigación Bibliográfica**

La investigación bibliográfica constituye una necesaria primera etapa de toda investigación, además esta proporciona el conocimiento de las investigaciones ya existentes –teorías, hipótesis, experimentos, resultados, instrumentos y técnicas usadas– acerca del tema o problema que el investigador se propone investigar o resolver.

En el caso específico de este trabajo de investigación, se utilizó la Investigación Bibliográfica para realizar un estudio del estado de las investigaciones realizadas hasta la fecha que se relacionaban de alguna forma con el objetivo de esta investigación, aportando de esta forma la base teórica indispensable sobre la cual se apoyaron los fundamentos de este trabajo.

11.2. Investigación Exploratoria

La Investigación Exploratoria será la primera fase que cumpla un investigador, sobre un objeto de estudio que resulte desconocido para él, o incluso también para el resto de la comunidad profesional del campo en el que se realice la investigación, careciendo entonces de antecedentes que puedan orientar la investigación emprendida.

Este método de investigación, en el diseño e implementación de sistema informático de gestión de información sobre la evaluación de desempeño en la Universidad Técnica de

Cotopaxi, ha sido abordado por él autor para fomentar el conocimiento de la importancia, viabilidad y los beneficios de las implementaciones que se han realizado a nivel nacional e internacional de sistemas similares al de esta investigación.

11.3.Métodos de investigación

11.3.1. Método inductivo

El método inductivo es aquel método científico que alcanza conclusiones generales partiendo de hipótesis o antecedentes en particular. Fuentes expresan que este método originalmente puede ser asociado a estudios de Francis Bacon a inicios del siglo XVII. El método inductivo suele basarse en la observación y la experimentación de hechos y acciones concretas para así poder llegar a una resolución o conclusión general sobre estos

A través de este métodos se efectuaron las conclusiones sobre los resultados obtenidos en las encuestas dirigidas al personal y la entrevista a la parte directiva de la institución; los cuales serán de gran utilidad para el diseño e implementación del sistema informático que ayudarán a generar mayor eficiencia en cada uno de los procesos en estudio.

11.3.2. Método deductivo

El método deductivo es un método científico que considera que la conclusión se halla implícita dentro las premisas. Esto quiere decir que las conclusiones son una consecuencia necesaria de las premisas: cuando las premisas resultan verdaderas y el razonamiento deductivo tiene validez, no hay forma de que la conclusión no sea verdadera

Este método se empleó para la deducción de la problematización, justificación, las preguntas científicas, conclusiones y recomendaciones generales en el desarrollo de la investigación.

11.3.3. Método analítico

El Método analítico es aquel método de investigación que consiste en la desmembración de un todo, descomponiéndolo en sus partes o elementos para observar las causas, la naturaleza y los efectos. El análisis es la observación y examen de un hecho en particular. Es necesario conocer la naturaleza del fenómeno y objeto que se estudia para comprender su esencia. Este método nos permite conocer más del objeto de estudio, con lo cual se puede: explicar, hacer analogías, comprender mejor su comportamiento y establecer nuevas teorías.

Para efectuar la investigación fue de vital importancia la utilización de este método en el análisis de la información teórica recopilada; así como el análisis de los resultados obtenidos mediante las encuestas y entrevista por tal razón el método analítico se considera una parte fundamental para el diseño e implantación de este sistema.

11.4. Técnicas de Investigación

11.4.1. Observación directa

Esta técnica consiste en observar atentamente el fenómeno, hecho o caso, tomar información y registrarla para su posterior análisis. La observación es un elemento fundamental de todo proceso investigativo; en ella se apoya el investigador para obtener el mayor número de datos.

En el desarrollo de esta investigación se aplicó esta técnica para lograr una mayor confiabilidad y asertividad en la información recopilada, recurriendo a la observación directa para la determinar la satisfacción del cliente con el sistema implementado.

12. Presupuesto para la elaboración del proyecto

| Resultados/Actividades | Trimestres | | | |
|---|-------------------|-----------------|-----------------|-------------------|
| | 1ro | 2do | 3ro | 4to |
| RECURSOS HUMANOS | | | | |
| Analista de programación | - | - | \$ 800,00 | \$ 800,00 |
| Analista de bases de datos | \$ 300,00 | - | - | - |
| Diseñador gráfico | - | \$ 100,00 | - | - |
| RECURSOS MATERIALES | | | | |
| Impresiones | \$ 12,50 | \$ 12,50 | \$ 12,50 | \$ 12,50 |
| Resma de hojas A4 | \$ 8,00 | - | - | - |
| Fotocopias de libros | \$ 5,00 | \$ 5,00 | - | - |
| Material de oficina | \$ 6,00 | - | - | - |
| Anillados | \$ 12,00 | - | \$ 12,00 | - |
| Encuadernación | - | - | - | \$ 75,00 |
| RECURSOS TECNOLÓGICOS | | | | |
| Internet | \$ 35,00 | \$ 35,00 | \$ 35,00 | \$ 35,00 |
| Servidor Core i5 | \$ 700,00 | - | - | - |
| Alquiler de Computadora de escritorio Core i5 | \$ 125,00 | \$ 125,00 | \$ 125,00 | \$ 125,00 |
| Microsoft Visual Studio 2012 | \$ 135,00 | - | - | - |
| Microsoft SQL Server 2008 | \$ 280,00 | - | - | - |
| CD | \$ 4,00 | - | - | - |
| Subtotal | \$1.339,50 | \$684,50 | \$984,50 | \$1.047,50 |
| Total | \$4.059,00 | | | |

Tabla 3. Presupuesto

Fuente: Equipo de Investigación

13. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

13.1. Muestreo aleatorio de la población

Una población estadística es un conjunto de sujetos o elementos que presentan características comunes. Sobre esta población se realiza el estudio estadístico con el fin de sacar conclusiones. (Escolares, 2014)

El tamaño poblacional es el número de individuos que constituyen la población. Según el número de sujetos, el tamaño puede ser finito o infinito. Los conjuntos infinitos son algo artificial o conceptual, ya que toda población de entidades físicas es finita, mientras que el muestreo es el proceso de seleccionar un conjunto finito de individuos de esta población con el fin de estudiarlos y poder caracterizar el total de la población. (Galois, 2015)

Para el caso de esta investigación se analizan dos poblaciones, la primera está formada por el conjunto de estudiantes de la Universidad Técnica de Cotopaxi la cual tiene un tamaño de 9000. La segunda población analizada sería los docentes de la misma institución, la cual tiene un tamaño de 360.

Partiendo de que las poblaciones analizadas siguen ambas una distribución normal se realiza el muestreo estadístico mediante la ecuación:

$$n = \frac{Z^2 \sigma^2 N}{e^2 (N - 1) Z^2 \sigma^2}$$

Donde hacemos las siguientes definiciones:

n: Tamaño de la muestra

N: Tamaño de la población, en este caso 9000 estudiantes

σ : Desviación estándar de la población, para la cual tomamos 0.5 tratándose de una distribución normal.

Z = Valor obtenido mediante niveles de confianza, para los cuales tenemos que:

1,96 implica un 95% de confianza equivale (como más usual)

2,58 implica un 99% de confianza.

Para el caso específico de esta investigación tomaremos 1,96 para un valor de confianza el 95%.

e: Límite aceptable de error muestral, este generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09). Para nuestro caso tomaremos un valor cercano a la media del error, lo cual sería 0,05. Sustituyendo en la ecuación anterior quedaría:

$$n = \frac{1.96^2 * 0.5^2 * 9000}{0.05^2 * (9000 - 1) * 1.96^2 * 0.5^2}$$

$$n = 51.02$$

Lo que redondeando daría un tamaño de muestra de 51 estudiantes. De esta misma manera calculamos el tamaño de la muestra de docentes, sustituyendo N=360 y e=0.09:

$$n = \frac{1.96^2 * 0.5^2 * 360}{0.09^2 * (360 - 1) * 1.96^2 * 0.5^2}$$

$$n \sim 16$$

Redondeando por exceso tendríamos una muestra de 16 docentes. De esta manera, con las anteriores muestras de las poblaciones de estudiantes y profesores de la institución se procedió a aplicar las encuestas detalladas en los anexos, de las cuales se puede inferir los resultados expuestos en el siguiente epígrafe.

13.2. Análisis de los resultados de las encuestas

13.2.1. Resultados de las encuestas aplicadas

Analizando los resultados de las encuestas aplicadas a estudiantes y docentes se llegó a las conclusiones que se muestran a continuación:

El 98.33% de los estudiantes, casi la totalidad, están de acuerdo con que se evalúe el desempeño de los docentes. Además, el 91.67% de los encuestados opina que no es adecuado que el docente se autoevalúe. Por otro lado, todos los estudiantes están de acuerdo con que los ellos puedan evaluar a los docentes que le impartieron materias en un ciclo que culmina.

Sólo el 16.67% de los estudiantes están en desacuerdo con que los docentes puedan evaluarse entre ellos. Mientras que, todos están de acuerdo en que los directivos evalúen a sus subordinados. Por otra parte, el 95 % de los estudiantes estuvieron de acuerdo con que el docente no pueda conocer el nombre, ni la puntuación recibida de la persona que le evaluó.

Con respecto al conocimiento sobre tecnología las encuestas arrojaron que el 88.33% de los encuestados, desconoce sobre los sistemas informáticos para de gestión de información, mientras que, 98.33% considera adecuado la implementación de un sistema informático para la gestión de las evaluaciones del personal docente. Además, sólo el 1.67% desconoce sobre los sistemas WEB, pero todos los encuestados están de acuerdo con la implementación de un sistema WEB que permita a los estudiantes, docentes y directivos la evaluación del personal docente.

Del análisis anterior se concluye que por la demanda entre los estudiantes y docentes de la Universidad técnica de Cotopaxi se considera necesario y viable la implementación de un sistema informático online que permita la evaluación del personal docente por parte de los estudiantes, directivos, otros docentes y el propio docente a ser evaluado.

13.3.Requerimientos funcionales

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, es el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Seguidamente se describen los requerimientos funcionales que son objetivos del este trabajo investigativo (Arias Chaves, 2005).

13.3.1. Requerimientos funcionales de proceso o área de negocio

1. El sistema permitirá a los estudiantes ingresar evaluaciones a los docentes.
2. El sistema permitirá a los docentes ingresar evaluaciones a otros docentes para la evaluación por pares.
3. El sistema permitirá a los directivos superiores evaluar el desempeño de los docentes bajo su cargo.
4. El sistema permitirá consultar por parte de los directivos la evaluación de los docentes bajo su cargo.

5. El sistema permitirá a los docentes consultar su evaluación total.
6. El sistema permitirá al administrador registrar nuevos usuarios en el sistema.
7. El sistema permitirá al administrador gestionar los indicadores a ser evaluados en los docentes.
8. El sistema permitirá al administrador ingresar el valor de ponderación para cada grupo de indicadores.
9. El sistema permitirá al administrador consultar todas las evaluaciones de todos los docentes y ordenarlos por cada campo.

13.3.2. Requerimientos funcionales de interfaz gráfica

1. El sistema validara automáticamente cada valor introducido.
2. El sistema mostrará la notificación correspondiente en caso de la entrada de un valor incorrecto.
3. Los campos para introducir las evaluaciones solo admiten valores decimales entre 0 y 10.
4. Para registrar nuevos usuarios, los campos nombres y apellidos aceptan caracteres alfabéticos únicamente.
5. Para registrar nuevos usuarios, el campo dirección acepta caracteres alfabéticos, numéricos y especiales.
6. Para registrar nuevos usuarios, el campo contraseña acepta caracteres alfabéticos, numéricos y especiales.
7. Para registrar nuevos usuarios, el campo contraseña no debe ser menor que 8.

13.3.3. Requerimientos funcionales legales o regulatorios

1. El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.

13.4. Requerimientos no funcionales

13.4.1. Requerimientos de Eficiencia

1. El sistema debe ser capaz de procesar 1000 evaluaciones por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al Software Testing de servicios web.

2. Toda funcionalidad del sistema debe responder al usuario en menos de 5 segundos.
3. El sistema debe ser capaz de operar adecuadamente con hasta 10.000 usuarios con sesiones concurrentes.
4. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

13.4.2. Requerimientos de Seguridad lógica y de datos

1. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
2. El sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
3. La Base de Datos debe respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura.
4. Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.

13.4.3. Requerimientos de Usabilidad

1. El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 10 horas.
2. La tasa de errores cometidos por el usuario deberá ser menor del 1% de las evaluaciones totales ejecutadas en el sistema.
3. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
4. La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
5. El sistema debe poseer interfaces gráficas bien formadas.

13.5. Sistema en funcionamiento

13.5.1. Pantalla principal

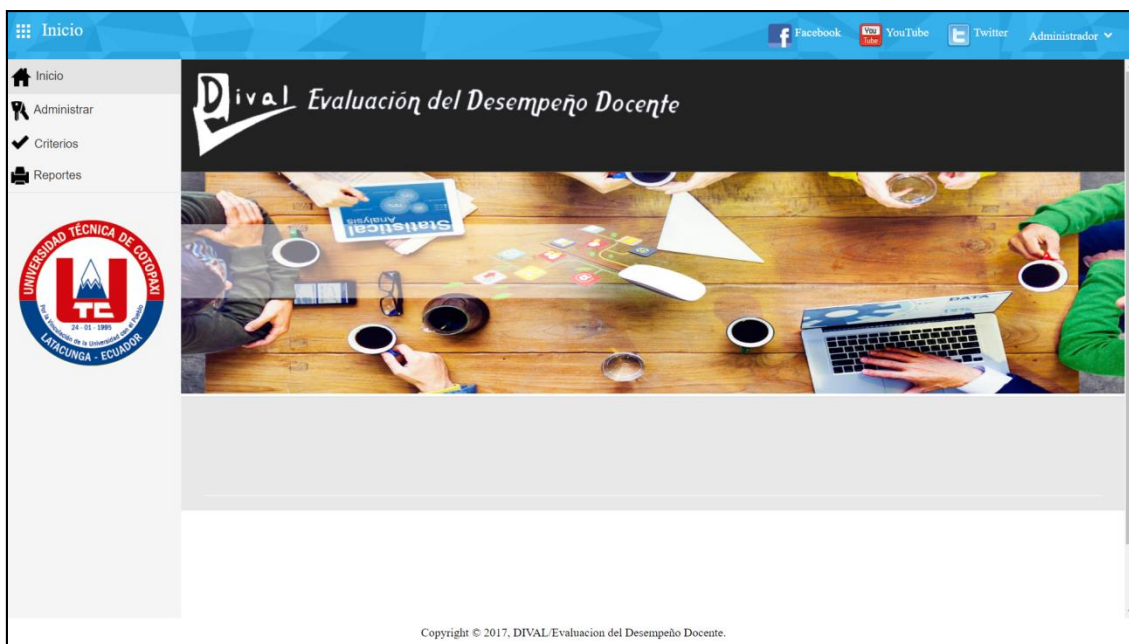


Figura 15. Pantalla principal

Fuente: Equipo de Investigación

En la Figura 15 se puede ver la pantalla principal de la aplicación y los elementos más significativos de la misma. En este apartado veremos todos estos elementos integrantes del entorno, de manera general y a medida que vayamos incursionando en el sistema los iremos viendo con detalle.

La llamamos pantalla principal porque, como veremos más adelante, existen otros entornos de trabajo que cambian completamente el aspecto de la misma, estos entornos distintos aparecen cuando trabajamos con las tablas que permiten la gestión de información.



Figura 16. Menú principal del administrador

Fuente: Equipo de Investigación

Tal como se indica en la Figura 15, en la parte superior aparecen horizontalmente la barra de Título (nombre de la aplicación), la barra de Menús a la izquierda (Figura 16) y tres botones para los contactos en las redes sociales. En la parte inferior se encuentra las credenciales de la institución. En el centro se encuentra el espacio más grande ocupado por una imagen decorativa relacionada con el propósito de la aplicación.

13.5.2. Gestión de roles

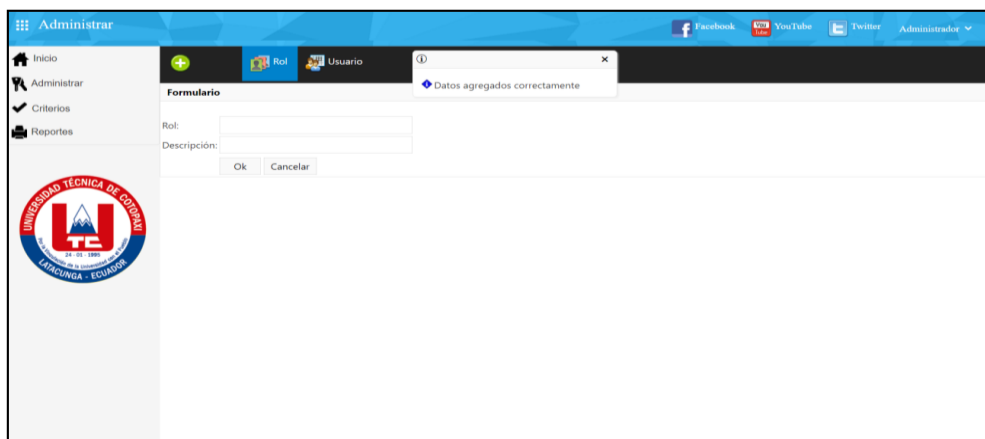


Figura 17. Gestión de roles

Fuente: Equipo de Investigación

Los usuarios se dividen en tres roles dentro del sistema: estudiantes, docentes y administrador. Los estudiantes solo tienen acceso para evaluar a los docentes que le impartieron clases en el ciclo académico concluyente, mientras que los docentes pueden evaluar al par que le ha sido asignado por el directivo, por su parte el administrador tiene todos los privilegios sobre el sistema, por ejemplo: gestiona los usuarios, roles, cargos, indicadores evaluables y sus pesos, directivos, etc. En la Figura 17 puede verse el módulo principal para la gestión de roles.

Figura 18. Nuevo rol

Fuente: Equipo de Investigación

En la Figura 18 puede verse la pantalla para agregar un nuevo rol al sistema. En este caso no figuran los permisos que tiene un rol sobre el sistema, debido a que estos se manejan directamente desde el código fuente de la aplicación lo que garantiza una mayor seguridad de la información almacenada.

13.5.3. Gestión de usuarios

En la Figura 19 puede verse la pantalla diseñada para agregar nuevos usuarios al sistema. En la misma aparecen los campos donde se introducen los datos generales del usuario como son: Nombre, Cédula, etc. Pero además tenemos algunos campos de vital importancia; El campo “Cargo” determina el cargo del docente dentro del sistema, por ejemplo: Director o Coordinador de Carrera, Este cargo determina si este docente además tiene que evaluar a sus subordinados o emparejar los pares de evaluación, como es el caso del coordinador de carrera. Además el cargo determina el porcentaje que representa cada grupo de indicadores en su evaluación. Por ejemplo: Para un director los indicadores de formación sólo representan el 30 % de su evaluación, mientras que para un docente investigador representa el 70 % de su evaluación. Además también aparece el campo de rol, donde se especifica uno de los roles gestionados con sus interfaces.

Figura 19. Gestión de usuarios

Fuente: Equipo de Investigación

13.5.4. Gestión de indicadores de evaluación

La gestión de los indicadores evaluativos comienza en una lista de estos como se muestra en la Figura 20, donde puede verse además los controles de editar y eliminar para cada elemento.



| No. | Indicador | Auto | Hetero | Par | Directivo | |
|-----|--|------|--------|-----|-----------|---|
| 0 | Elaboración, actualización y/o perfeccionamiento del sílabo | 1 | 1 | 1 | 2 |   |
| 0 | Impartición de clases | 1 | 3 | 2 | 1 |   |
| 0 | Elaboración de manuales para la docencia, repositorios y/o guías | 1 | 2,5 | 1 | 1 |   |
| 0 | Uso de materiales didácticos reales o virtuales | 1 | 3 | 2 | 0,5 |   |

Figura 20. Lista de indicadores evaluativos

Fuente: Equipo de Investigación

Otra interfaz que forma parte de este proceso es la mostrada en la Figura 21 se puede observar los campos requeridos por cada indicador. En la parte inferior aparecen 4 campos con los nombres: Auto, Hétero, Par y Directivo. Esos campos guardan número reales, los cuales indican los pesos de estos indicadores dentro de cada canal de evaluación. Estos canales de evaluación se describen como:

1. Auto: Es la evaluación realizada por el propio docente sobre su desempeño.
2. Hétero: Se califica como la media aritmética de las evaluaciones recibidas por el docente de todos los estudiantes a los que este les dio clases en un ciclo.
3. Par: Es la evaluación recibida por el docente de parte del par asignado por el coordinador de carrera.
4. Directivo: Es la evaluación recibida por parte del jefe inmediato superior del docente.

Figura 21. Gestión de indicadores evaluativos

Fuente: Equipo de Investigación

13.6. Pruebas de Caja Blanca y Caja Negra

13.6.1. Error de conversión de tipos incompatibles

En un lenguaje estáticamente tipado como C#, el cual fue el seleccionado para el desarrollo del sistema, cada variable debe ser declarada con un tipo. Para utilizar una variable de tipo entero, es necesario indicar que en efecto, es de ese tipo. El tipado estático es típico de los lenguajes compilados. El tipo no se altera hasta que la variable se sale de ámbito y se extingue. Éste comportamiento se aplica tanto a variables locales (a un método o función), a parámetros (de un método o función), a variables de instancia (de un objeto), a variables de clase ("static"), o a variables globales (en lenguajes no orientados a objetos).

En el transcurso del desarrollo de nuestra aplicación se produjeron varias veces errores de conversión de tipos incompatibles como se muestra en la Figura 22.

La solución a este error estuvo en definir una clase con métodos especiales que se pueden para habilitar las conversiones explícitas e implícitas entre tipos personalizados que no tienen una relación de clase base-clase derivada.

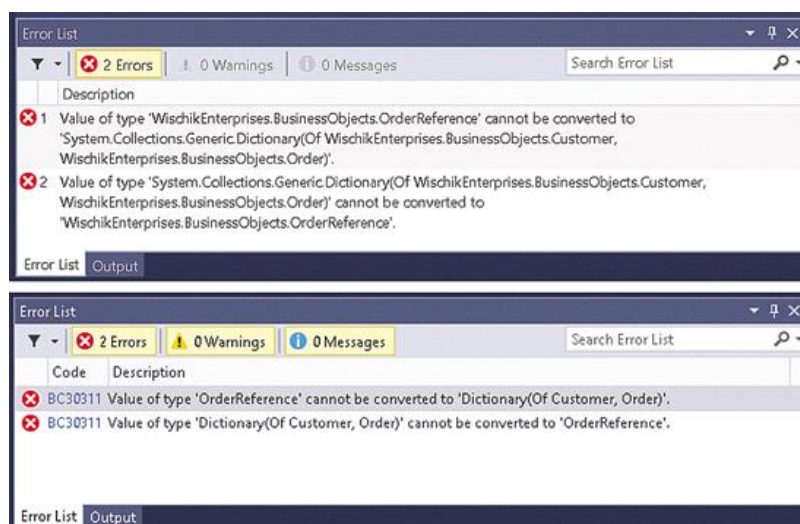


Figura 22. Error de conversión de tipos

Fuente: Equipo de Investigación

13.6.2. Error de acceso a archivos ensamblados

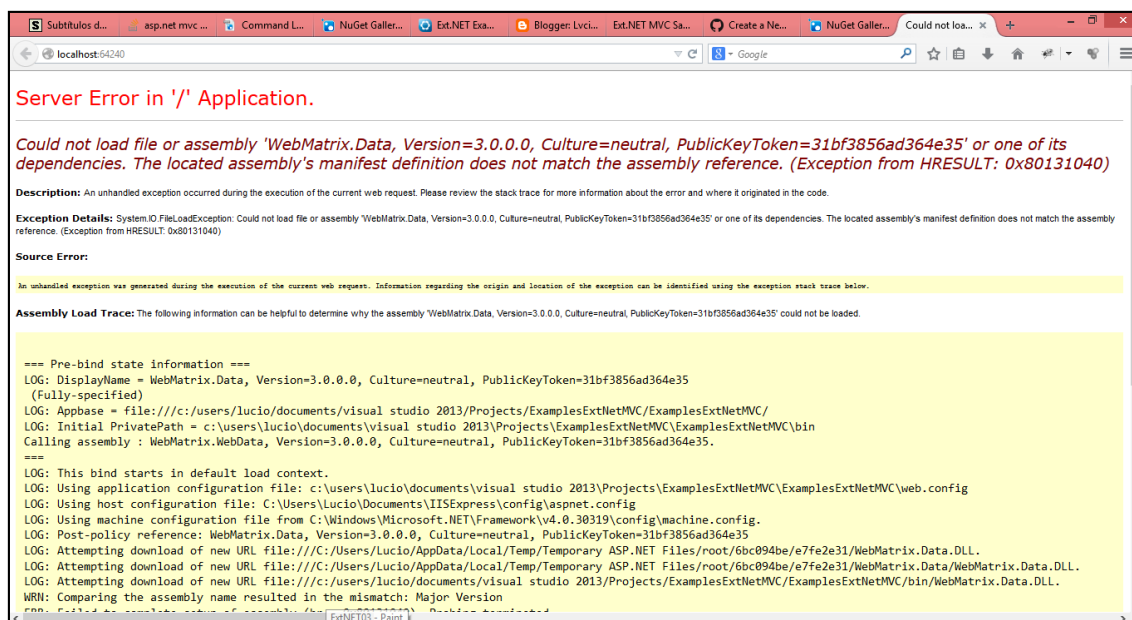


Figura 23. Error de acceso a archivos ensamblado

Fuente: Equipo de Investigación

El error que se describe a continuación sucedió en luego del sistema estar compilado y desplegado, en el momento de ejecutar la pantalla principal del sistema. La Figura 23 es una captura del error presentado en el momento de la primera ejecución de la aplicación desplegada en un servidor virtual.

Este error se describe como un acceso incorrecto a los archivos ensamblados que conforma lo que sería el sistema de archivos de la aplicación. Una vez comprobado la existencia del archivo y gracias a la investigación en los foros de discusión sobre el tema se comprendió que el problema no estaba dado por la no existencia de los archivos, si no, por los permisos de estos. La solución fue otorgar permisos de lectura/escritura al proceso de Internet Information Server

14. CONCLUSIONES

- Se capturaron los requerimientos funcionales para el sistema por medio de entrevistas con los directivos de la Entidad.
- Se analizaron las herramientas disponibles en la actualidad para el desarrollo de sistemas informáticos y se seleccionaron las adecuadas para el desarrollo del sistema.
- Se implementó el sistema informático aplicando las herramientas adecuadas consiguiendo un desarrollo eficiente y cumpliendo los plazos de entrega.

15. RECOMENDACIONES

- Si surge la necesidad de la adición de nuevas funcionalidades al sistema, se recomienda que se implemente de manera modular manteniendo la conexión con los módulos actuales sin la necesidad de modificar estos.
- Para la inclusión de nuevos módulos en el sistema se recomienda que estos sean implementados con las mismas herramientas que se usaron para el desarrollo del sistema desde su fase inicial.
- Si aparece la necesidad de modificar alguno de los módulos actuales, se recomienda que se usen las mismas herramientas y versiones usadas para el desarrollo del sistema.
- Se recomienda evitar las modificaciones directas en la base de datos debido a que esto puede ocasionar inconsistencia en la información almacenada y el mal funcionamiento de los módulos del sistema.

16. Bibliografía

Agrawal, S., & Gupta, R. D. (2014). Development and comparison of open source based Web GIS Frameworks on WAMP and Apache Tomcat Web Servers. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences .

Agrawal, S., Chaudhuri, S., & Kollar, L. (2005). Database tuning advisor for microsoft SQL server 2005: demo. En S. Agrawal, Database tuning advisor for microsoft SQL server 2005: demo (págs. 930-932). ACM.

Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. InterSedes: Revista de las Sedes Regionales .

Böck, H. (2011). The Definitive Guide to NetBeans Platform 7. Apress .

Boudreau, T. a., Greene, S., Spurlin, V., & Woehr, J. J. (2012). NetBeans: the definitive guide. O'Reilly Media, Inc.

Briggs, O., Champeon, S., Costello, E., & Patterson, M. (2013). Cascading Style Sheets.

Buffington, J. (2010). Microsoft sql server. Data Protection for Virtual Data Centers , 267-315.

Canós, J., Letelier, P., & Penadés, C. (2013). Metodologías ágiles en el desarrollo de software. Metodologías Ágiles en el Desarrollo de Software , 11.

Castagnetto, J. (2015). Professional PHP programming. Apress.

Catalani, E. (2014). Arquitectura Modelo/Vista/Controlador. Extraí2 .

Cecilia Rodríguez, J. (2015). Desarrollo de una red social de futbol que incorpora la creación de ligas virtuales entre usuarios, usando el framework Symfony 2.

Christopher, A. (2012). CSS Adopte las hojas de estilo para dominar los estándares de la web. España. Editorial ENI .

Coad, P., Luca, J., & Lefebvre, E. (2011). Java modeling color with UML: Enterprise components and process with Cdrom. Prentice Hall PTR.

Cobo, A. (2015). PHP y MySQL: Tecnología para el desarrollo de aplicaciones web. Ediciones Díaz de Santos.

Cockburn, A. (2016). Agile software development: the cooperative game. Pearson Education , 1.

Connolly, T., & Begg, C. (2015). Sistemas de bases de datos. Un enfoque práctico .

Consortium, W. W., & others. (2011). World Wide Web. World Wide Web Consortium , 1-10.

Crockford, D. (2016). JavaScript: The Good Parts: The Good Parts. O'Reilly Media, Inc.

Date, C. J., Faudón, S. L., & Ruiz, M. (2011). Introducción a los sistemas de bases de datos. Pearson Educación.

Diógenes, M. (2016). Librosweb. Obtenido de http://librosweb.es/libro/xhtml/capitulo_3.html

Eguiluz, J. (2013). Desarrollo web ágil con MS Visual Studio. España: Gestor de publicaciones.

Elmasri, R., Navathe, S. B., & Castillo. (2012). Fundamentos de sistemas de bases de datos. Addison-Wesley .

Escolares. (2014). Escolares.Net. Obtenido de <http://www.escolares.net/matematicas/poblacion-y-muestra-estadistica/>

Flanagan, D. (2015). JavaScript. La Guía Definitiva.

Galois, J. (2015). universoformulas. Obtenido de <http://www.universoformulas.com/estadistica/descriptiva/poblacion-estadistica/>

Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y Javascript. Marcombo.

Gómez, O. T., López, P. P., & Bacalla, J. S. (2011). Criterios de selección de metodologías de desarrollo de software. Industrial Data , 70-74.

González, Y. D., & Romero, Y. F. (2012). Patrón Modelo-Vista-Controlador. Revista Telemática , 47-57.

Gutiérrez, J. J. (2014). ¿Qué es un framework Web? Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf .

Hartati, G. S. (2016). Pemrograman GUI Swing Java dengan NetBeans 5. Andi Offset, Yogyakarta .

Highsmith, J. (2013). Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley.

Jeffries, R., Anderson, A., & Hendrickson, C. (2011). Extreme programming installed. Addison-Wesley Professional.

Joyanes Aguilar, L. (2014). Fundamentos de programación. Algoritmos, estructuras de datos y objetos. McGrawHill.

Joyanes Aguilar, L. (1996). Programación orientada a objetos. España: McGraw-Hill.

Korth, H. F., & Silberschatz, A. (2013). Fundamentos de bases de datos. McGraw-Hill.

Letelier, P. (2016). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Técnica Administrativa .

Msdn.microsoft.com. (13 de 7 de 2017). Información general sobre ASP.NET.

Newkirk, J., & Martin, R. C. (2015). Addendum to the 2000 proceedings of the conference on Object-oriented programming, systems, languages, and applications (Addendum). Extreme programming in practice (págs. 25-26). ACM.

Ojeda, F. C. (2015). La biblia de HTML. Anaya Multimedia.

Pérez López, C. (2016). Oracle 10g: administración y análisis de bases de datos. Yell's Science.

Petri, J. (2011). Developer RAI Platforms. Developer's Guide. Packt Publishing Ltd .

Poppendieck, M., & Poppendieck, T. (2013). Lean Software Development: An Agile Toolkit: An Agile Toolkit. Addison-Wesley.

Powell, T. A. (2011). HTML: manual de referencia. McGraw-Hill Interamericana de España.

Pressman, R. S. (2004). Ingeniería del software. McGraw Hill.

Prosky, L., Nils-Georg, Furda, I., & DeVries, J. (2004). Desarrollo de Software mediante plataformas RAID. Journal-Association of Official Analytical Software , 677-679.

Puertas, J. P. (2015). Navegar en Internet: creación de un portal con PHP y MySQL. Alfaomega.

Rodríguez, J. (2012). Sublime Text 2. Macworld España: la publicación para los usuarios del Mac, el iPhone y el iPod .

Schmitt, C. (2013). Curso de CSS.

Schwaber, K., & Beedle, M. (2012). Agile Software Development with Scrum. Prentice Hall , 3.

Silva, D., & Mercerat, B. (2011). Construyendo aplicaciones web con una metodología de diseño orientada a objetos. Revista Colombiana de Computación--RCC .

Skinner, J. (2015). Sublime text: The text editor you'll fall in love with. Sublime Text .

Stapleton, J. (2013). DSDM: Dynamic systems development method. En J. Stapleton, Technology of Object-Oriented Languages and Systems, 1999. Proceedings of (págs. 406-406).

Stothard, P. (2011). The sequence manipulation suite: JavaScript programs for analyzing and formatting protein and DNA sequences. Biotechniques , 1102-1104.

Veloz, H. (2010). Encuentro Iberoamericano sobre evaluación del desempeño docente. Ponencia presentada por Report'sPaskumr , 3.

Vértice, E. (2015). Diseño básico de páginas web en HTML. Editorial Vértice.

Vilas Usó, P. (2015). Desarrollo de una web corporativa basada en Plataformas RAID. Universitat Jaume I.

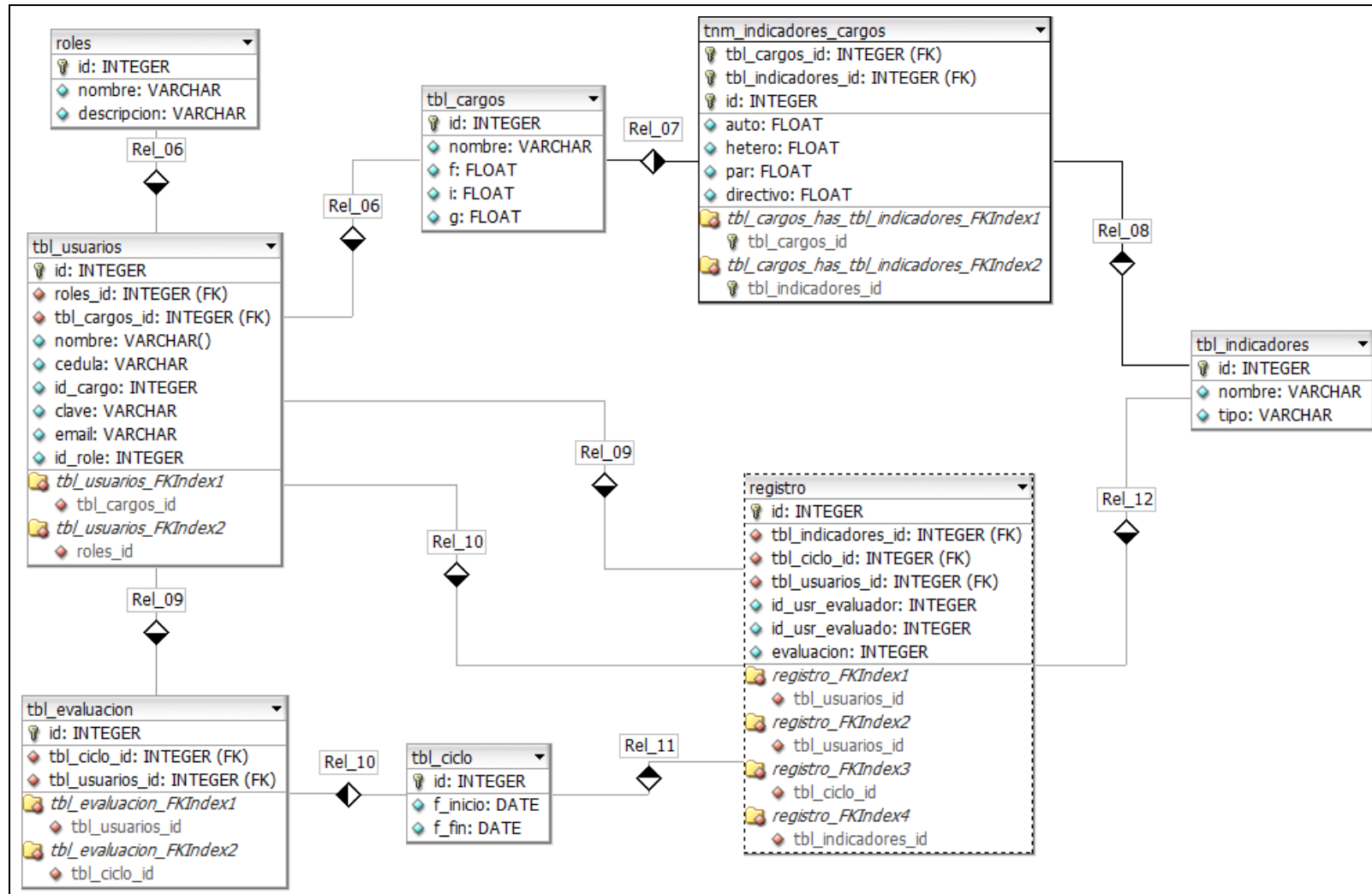
Welling, L., & Thomson, L. (2014). Desarrollo web con PHP y MySQL.

Zawodny, J. D., & Balling, D. J. (2014). MySQL avanzado.

Zubillaga Beraza, C. (2016). Migración de inmobiliaria a Joomla. Antik@. LTD .

17. Anexos

17.1. Anexo 1: Modelo relacional de la Base de Datos



17.2. Anexo 2: Script SQL de la base de datos

```

CREATE TABLE registro (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  tbl_indicadores_id INTEGER UNSIGNED NOT NULL,
  tbl_ciclo_id INTEGER UNSIGNED NOT NULL,
  tbl_usuarios_id INTEGER UNSIGNED NOT NULL,
  id_usr_evaluador INTEGER UNSIGNED NULL,
  id_usr_evaluado INTEGER UNSIGNED NULL,
  PRIMARY KEY(id),
  INDEX registro_FKIndex1(tbl_usuarios_id),
  INDEX registro_FKIndex2(tbl_usuarios_id),
  INDEX registro_FKIndex3(tbl_ciclo_id),
  INDEX registro_FKIndex4(tbl_indicadores_id)
);

CREATE TABLE roles (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nombre VARCHAR NULL,
  descripcion VARCHAR NULL,
  PRIMARY KEY(id)
);

CREATE TABLE tbl_cargos (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nombre VARCHAR NULL,
  f FLOAT NULL,
  i FLOAT NULL,
  g FLOAT NULL,
  PRIMARY KEY(id)
);

CREATE TABLE tbl_ciclo (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  f_inicio DATE NULL,
  f_fin DATE NULL,
  PRIMARY KEY(id)
);

CREATE TABLE tbl_evaluacion (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  tbl_ciclo_id INTEGER UNSIGNED NOT NULL,
  tbl_usuarios_id INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  INDEX tbl_evaluacion_FKIndex1(tbl_usuarios_id),
  INDEX tbl_evaluacion_FKIndex2(tbl_ciclo_id)
);

```

```

CREATE TABLE tbl_indicadores (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nombre VARCHAR NULL,
  tipo VARCHAR NULL,
  PRIMARY KEY(id)
);


CREATE TABLE tbl_usuarios (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  roles_id INTEGER UNSIGNED NOT NULL,
  tbl_cargos_id INTEGER UNSIGNED NOT NULL,
  nombre VARCHAR() NULL,
  cedula VARCHAR NULL,
  id_cargo INTEGER UNSIGNED NULL,
  clave VARCHAR NULL,
  email VARCHAR NULL,
  id_role INTEGER UNSIGNED NULL,
  PRIMARY KEY(id),
  INDEX tbl_usuarios_FKIndex1(tbl_cargos_id),
  INDEX tbl_usuarios_FKIndex2(roles_id)
);

CREATE TABLE tnm_indicadores_cargos (
  tbl_cargos_id INTEGER UNSIGNED NOT NULL,
  tbl_indicadores_id INTEGER UNSIGNED NOT NULL,
  id INTEGER UNSIGNED NOT NULL,
  auto FLOAT NULL,
  hetero FLOAT NULL,
  par FLOAT NULL,
  directivo FLOAT NULL,
  PRIMARY KEY(tbl_cargos_id, tbl_indicadores_id, id),
  INDEX tbl_cargos_has_tbl_indicadores_FKIndex1(tbl_cargos_id),
  INDEX tbl_cargos_has_tbl_indicadores_FKIndex2(tbl_indicadores_id)
);

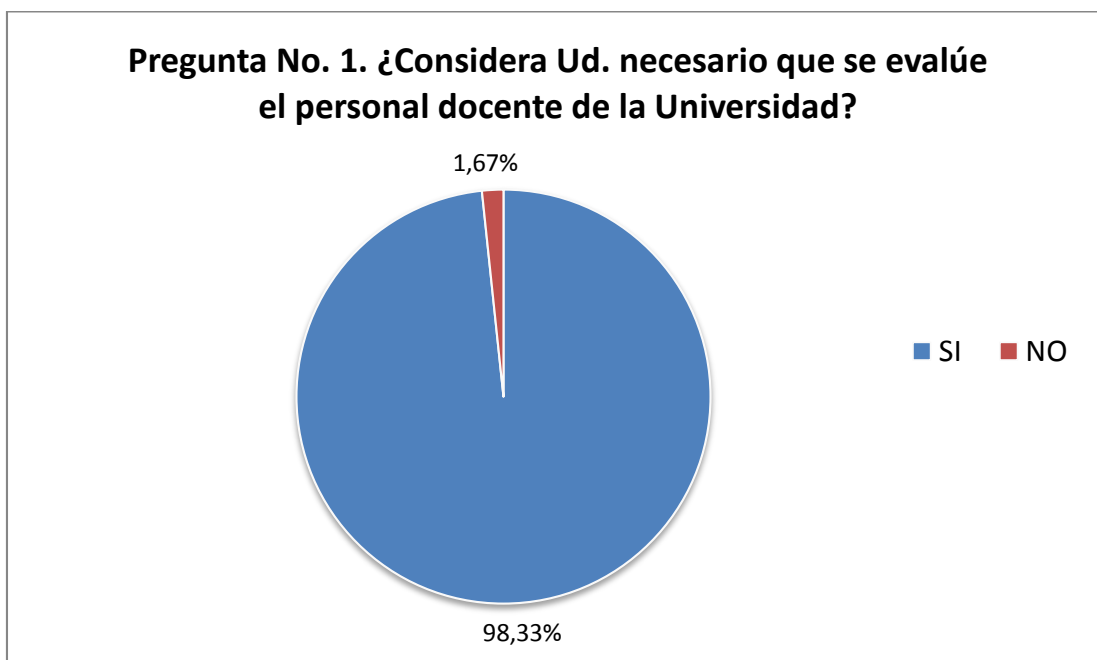
```

17.3. Anexo 3: Encuesta realizada a los estudiantes

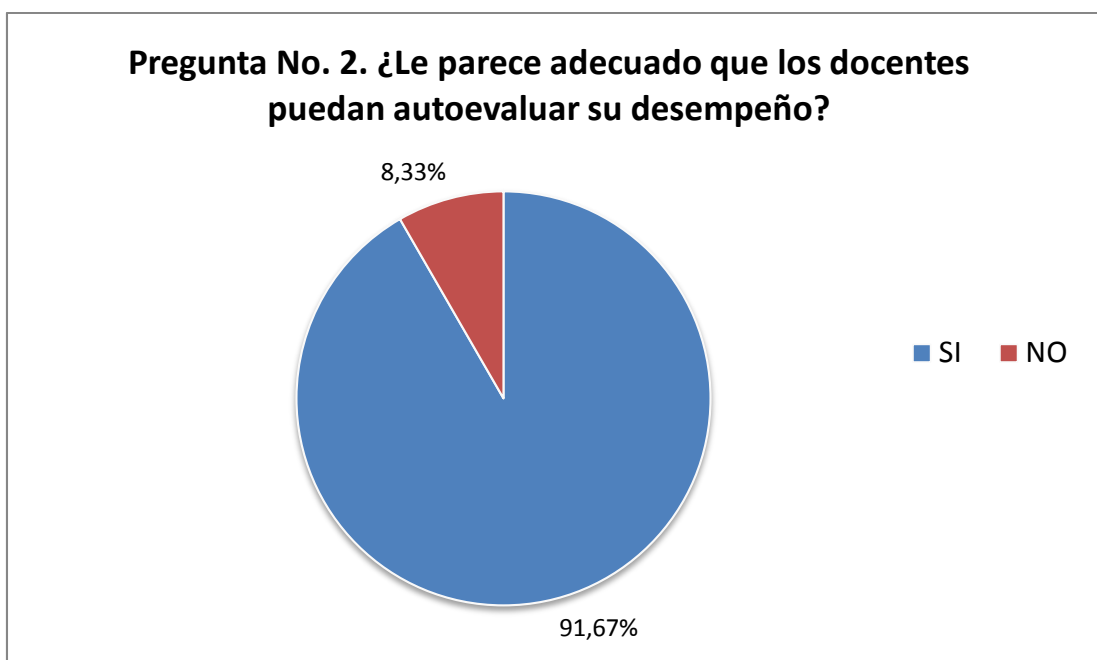
La siguiente encuesta fue realizada a una muestra de 60 estudiantes de la Universidad Técnica de Cotopaxi – Extensión La Maná con el fin de recaudar información necesaria para determinar la necesidad del presente proyecto.

|  <p>UNIVERSIDAD TÉCNICA DE COTOPAXI LA MANÁ ENCUESTA PARA PROYECTO DE INVESTIGACIÓN</p> | | |
|---|--|-------------------------------------|
| <p>Para efectos de la realización de este proyecto SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI se requiere recabar información para lo cual necesitamos conocer su opinión, por tal razón le agradecemos se digne contestar la siguiente encuesta.</p> | | |
| No. | PREGUNTA | RESPUESTA POSIBLES SI NO |
| 1. | ¿Considera Ud. necesario que se evalúe el personal docente de la Universidad? | |
| 2. | ¿Le parece adecuado que los docentes puedan autoevaluar su desempeño? | |
| 3. | ¿Le parece adecuado que los estudiantes evalúen a los docentes que le impartieron clases en el ciclo concluyente? | |
| 4. | ¿Considera necesario que los docentes puedan además evaluarse entre ellos en pares anónimos? | |
| 5. | ¿Considera necesario que los directivos puedan evaluar a los docentes que están bajo su cargo? | |
| 6. | ¿Considera necesario que el docente pueda conocer el nombre de cada persona que le evaluó y la puntuación que le dio? | |
| 7. | ¿Conoce sobre los sistemas informáticos para la gestión de información? | |
| 8. | ¿Considera necesario que se realice un sistema informático que permita gestionar la evaluación del personal docente teniendo en cuenta las evaluaciones por los distintos canales? | |
| 9. | ¿Conoce sobre los sistemas WEB o Online? | |
| 10. | ¿Considera necesario que la Universidad implemente un sistema informático en la WEB que permita a los estudiantes, docentes y directivos realizar las evaluaciones correspondientes? | |

17.4. Anexo 4: Resultados de la encuesta aplicada a los estudiantes

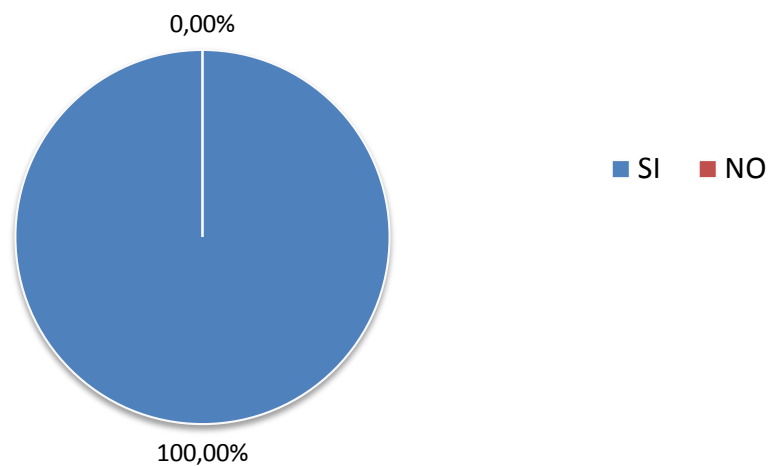


Fuente: Equipo de Investigación



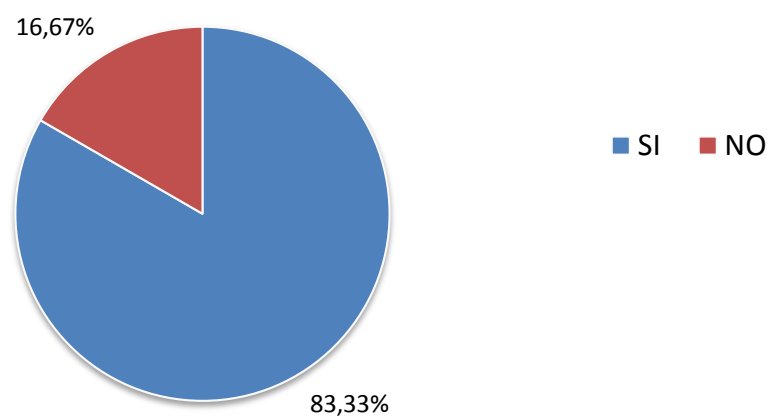
Fuente: Equipo de Investigación

Pregunta No. 3. ¿Le parece adecuado que los estudiantes evalúen a los docentes que le impartieron clases en el ciclo concluyente?



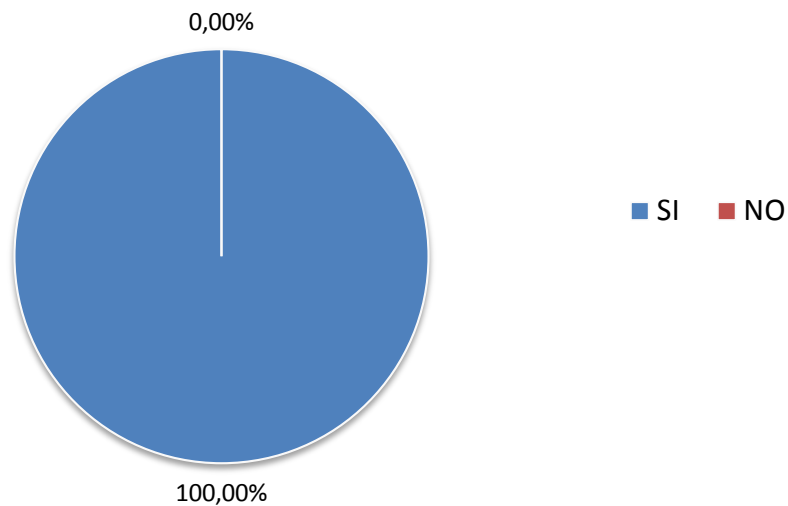
Fuente: Equipo de Investigación

Pregunta No. 4. ¿Considera necesario que los docentes puedan además evaluarse entre ellos en pares anónimos?



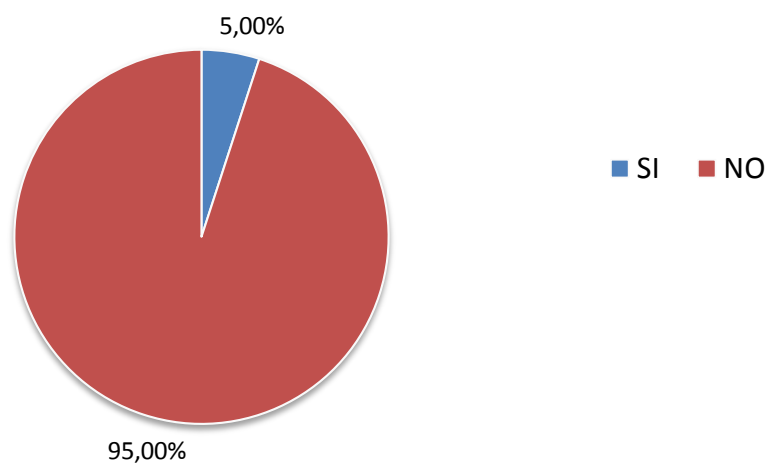
Fuente: Equipo de Investigación

Pregunta No. 5. ¿Considera necesario que los directivos puedan evaluar a los docentes que están bajo su cargo?



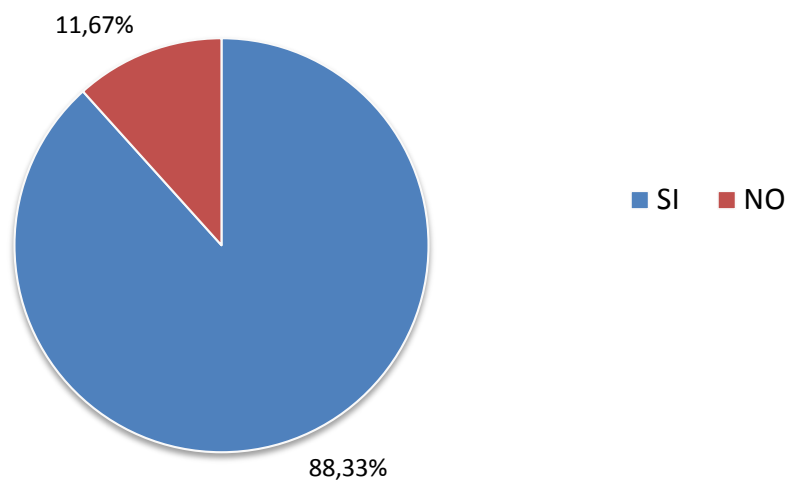
Fuente: Equipo de Investigación

Pregunta No. 6. ¿Considera necesario que el docente pueda conocer el nombre de cada persona que le evaluó y la puntuación que le dio?



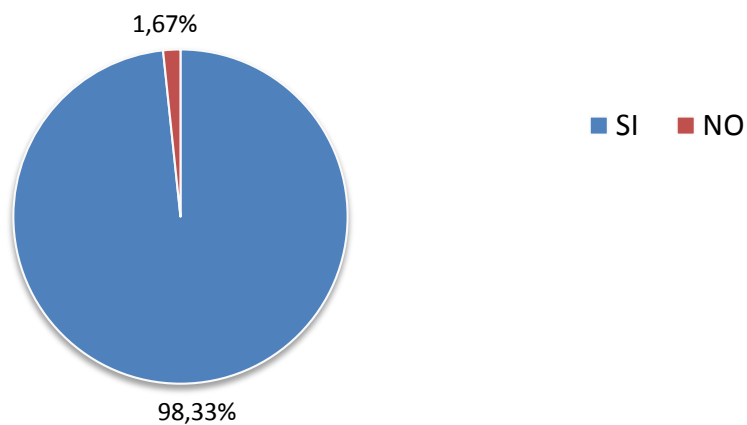
Fuente: Equipo de Investigación

Pregunta No. 7. ¿Conoce sobre los sistemas informáticos para la gestión de información?

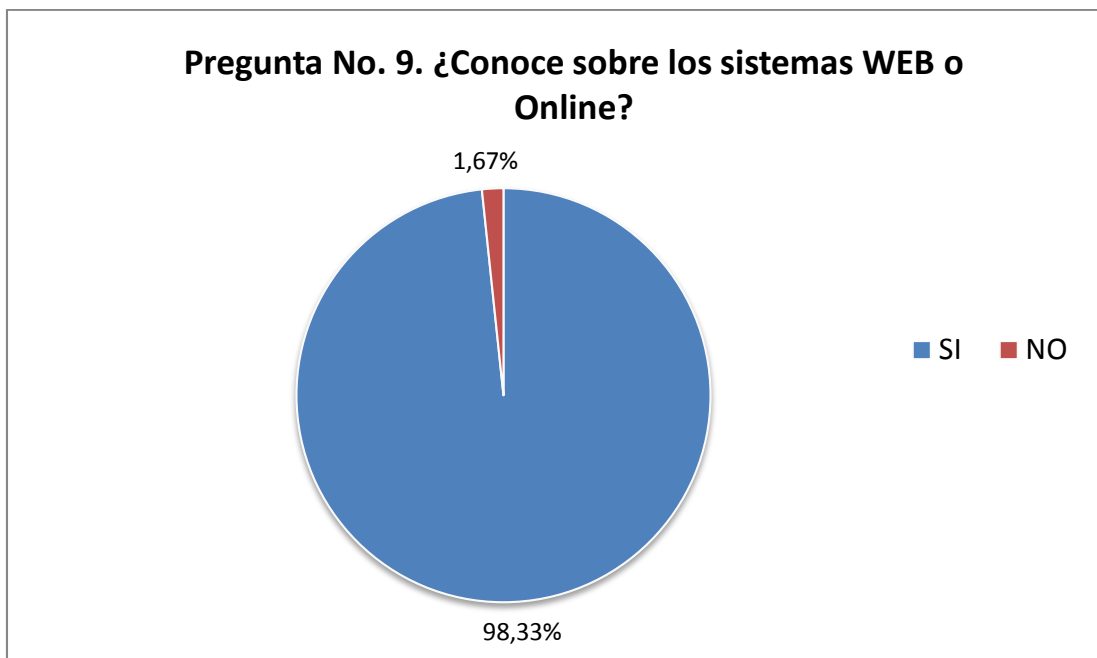


Fuente: Equipo de Investigación

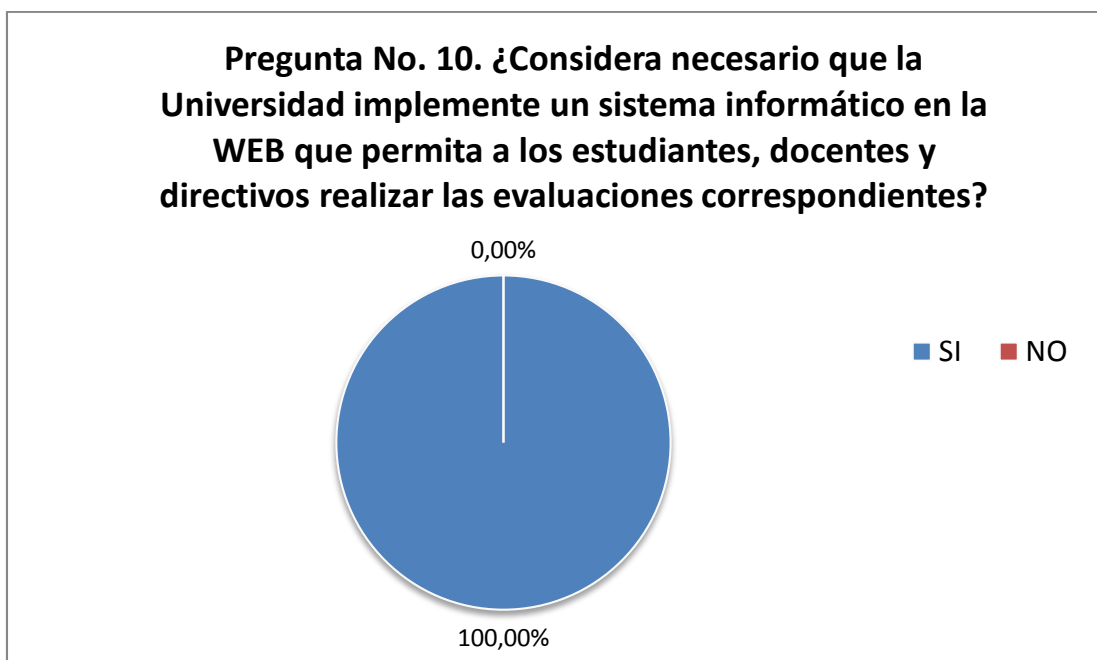
Pregunta No. 8. ¿Considera necesario que se realice un sistema informático que permita gestionar la evaluación del personal docente teniendo en cuenta las evaluaciones por los distintos canales?



Fuente: Equipo de Investigación




Fuente: Equipo de Investigación



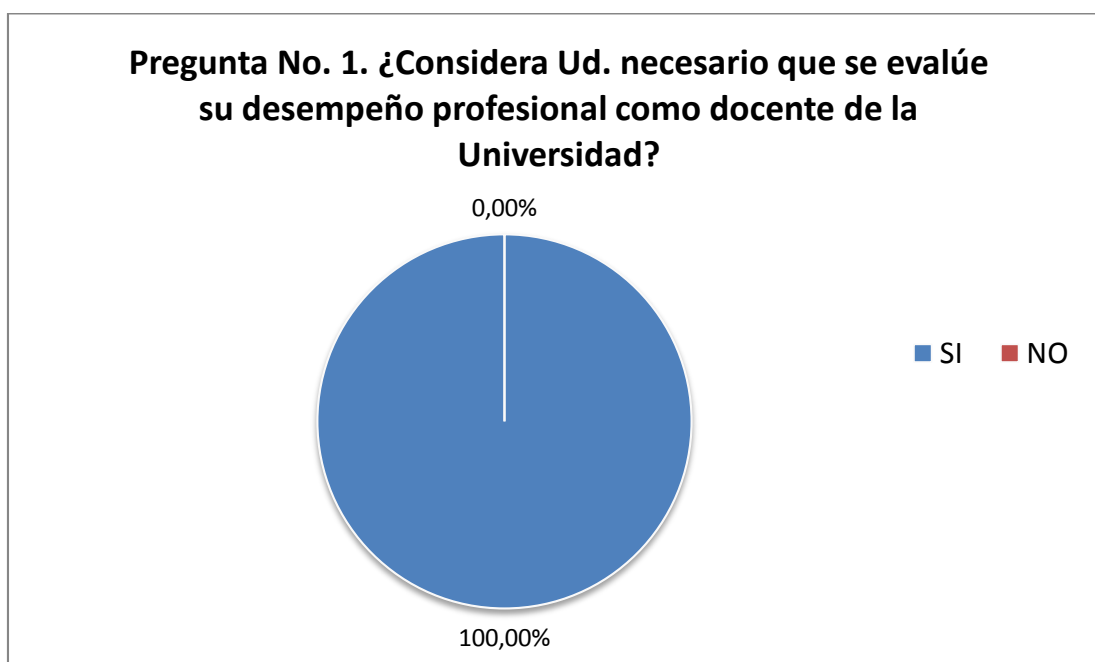
Fuente: Equipo de Investigación

17.5. Anexo 4. Encuesta realizada a los docentes

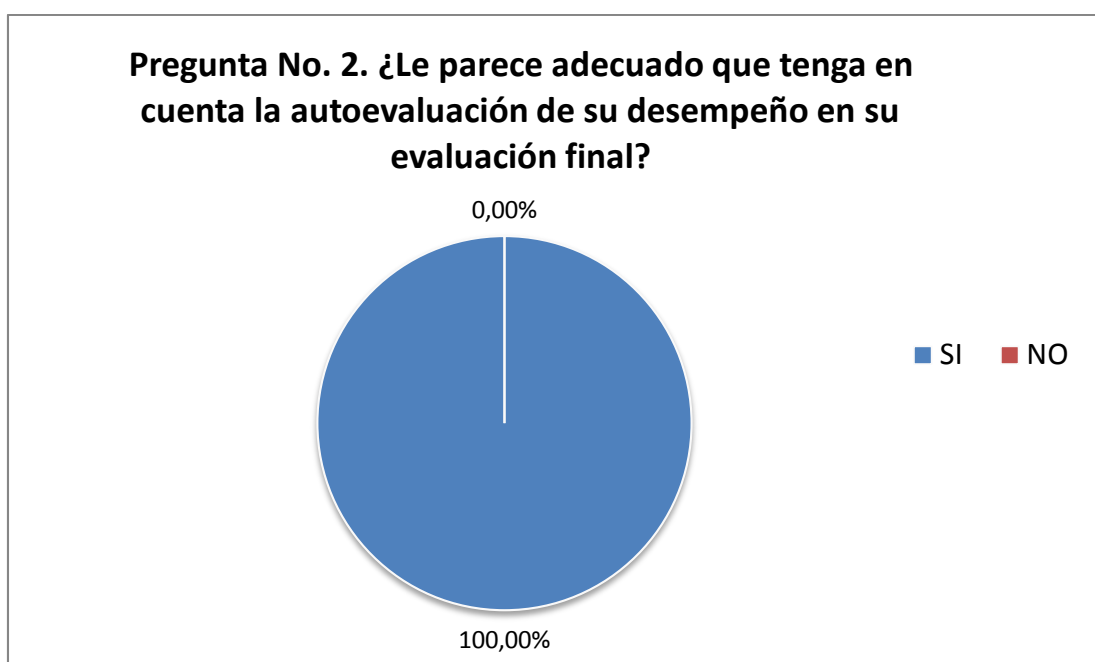
La siguiente encuesta fue realizada a una muestra de 10 docentes de la Universidad Técnica de Cotopaxi – Extensión La Maná con el fin de recaudar información necesaria para determinar la necesidad del presente proyecto.

|  <p>UNIVERSIDAD TÉCNICA DE COTOPAXI LA MANÁ ENCUESTA PARA PROYECTO DE INVESTIGACIÓN</p> | | |
|---|--|-------------------------------------|
| <p>Para efectos de la realización de este proyecto SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LA EVALUACIÓN DE DESEMPEÑO DE LOS DOCENTES EN LA UNIVERSIDAD TÉCNICA DE COTOPAXI se requiere recabar información para lo cual necesitamos conocer su opinión, por tal razón le agradecemos se digne contestar la siguiente encuesta.</p> | | |
| No. | PREGUNTA | RESPUESTA POSIBLES SI NO |
| 1 | ¿Considera Ud. necesario que se evalúe su desempeño profesional como docente de la Universidad? | |
| 2. | ¿Le parece adecuado que tenga en cuenta la autoevaluación de su desempeño en su evaluación final? | |
| 3. | ¿Le parece adecuado que sus estudiantes evalúen su desempeño como docente en el ciclo concluyente? | |
| 4. | ¿Considera necesario que otro docente compañero suyo pueda evaluar su desempeño y al mismo tiempo Ud. evaluar el de él? | |
| 5. | ¿Considera necesario que su jefe inmediato superior evalúe su desempeño profesional? | |
| 6. | ¿Considera que debe conocer el nombre de cada persona que le evaluó y la puntuación que le dio? | |
| 7. | ¿Conoce sobre los sistemas informáticos para la gestión de información? | |
| 8. | ¿Considera necesario que se realice un sistema informático que permita gestionar la evaluación del personal docente teniendo en cuenta las evaluaciones por los distintos canales? | |
| 9. | ¿Conoce sobre los sistemas WEB o Online? | |
| 10. | ¿Considera necesario que la Universidad implemente un sistema informático en la WEB que permita gestionar sus evaluaciones en cada ciclo académico? | |

17.6. Anexo 5: Resultados de la encuesta aplicada a los docentes

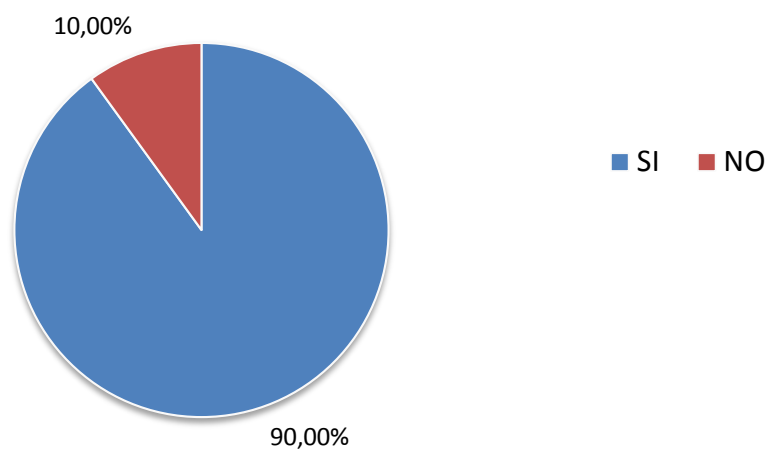


Fuente: Equipo de Investigación



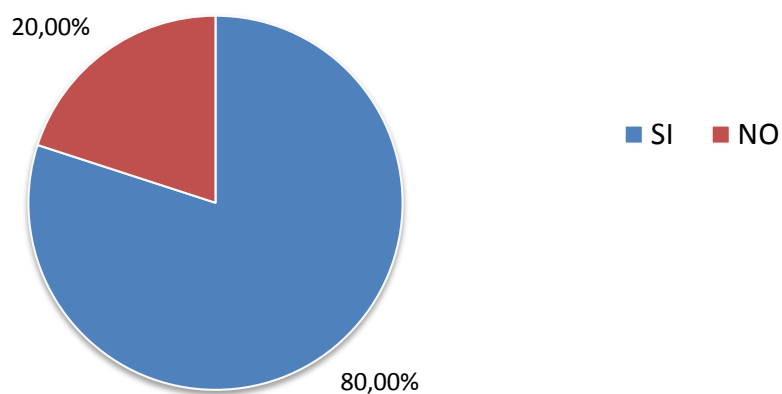
Fuente: Equipo de Investigación

Pregunta No. 3. ¿Le parece adecuado que sus estudiantes evalúen su desempeño como docente en el ciclo concluyente?



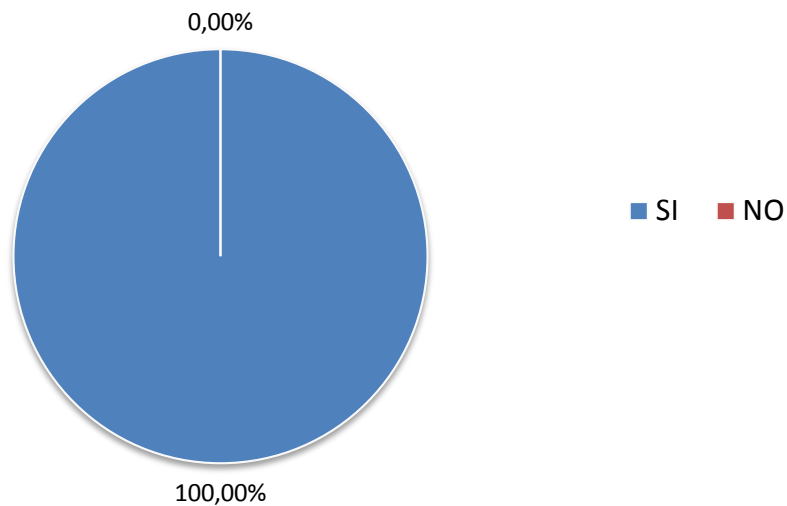
Fuente: Equipo de Investigación

Pregunta No. 4. ¿Considera necesario que otro docente compañero suyo pueda evaluar su desempeño y al mismo tiempo Ud. evaluar el de él?



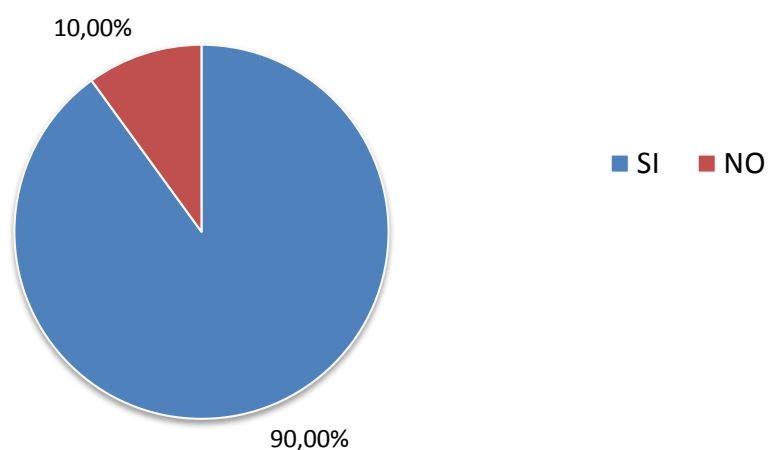
Fuente: Equipo de Investigación

Pregunta No. 5. ¿Considera necesario que su jefe inmediato superior evalúe su desempeño profesional?



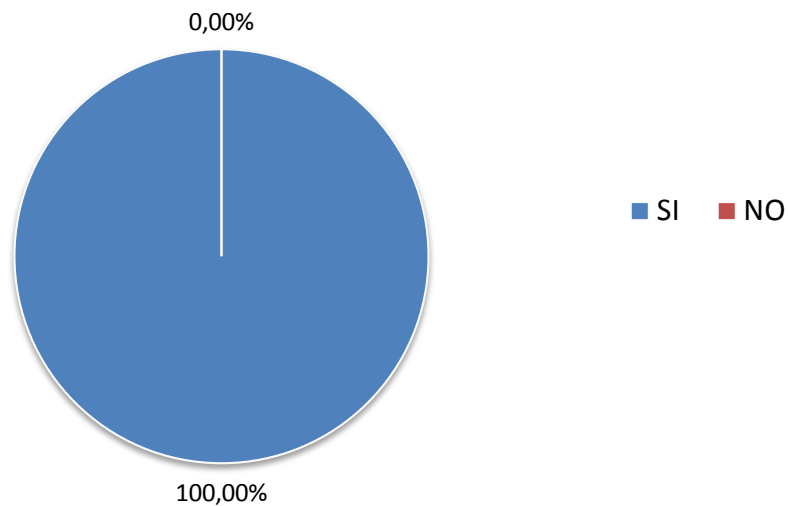
Fuente: Equipo de Investigación

Pregunta No. 6. ¿Considera que debe conocer el nombre de cada persona que le evaluó y la puntuación que le dio?



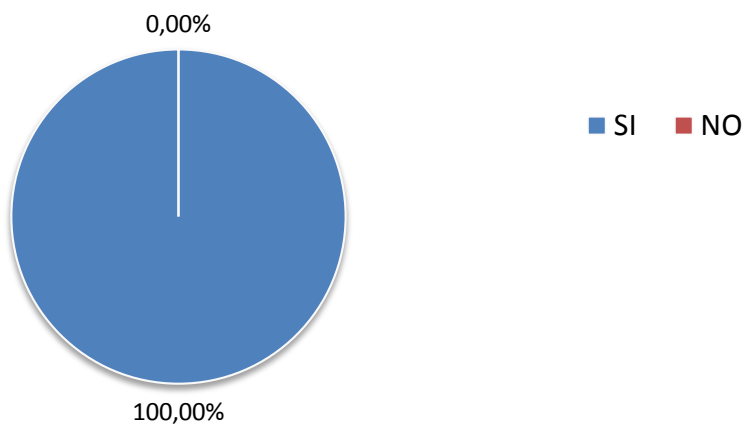
Fuente: Equipo de Investigación

Pregunta No. 7. ¿Conoce sobre los sistemas informáticos para la gestión de información?

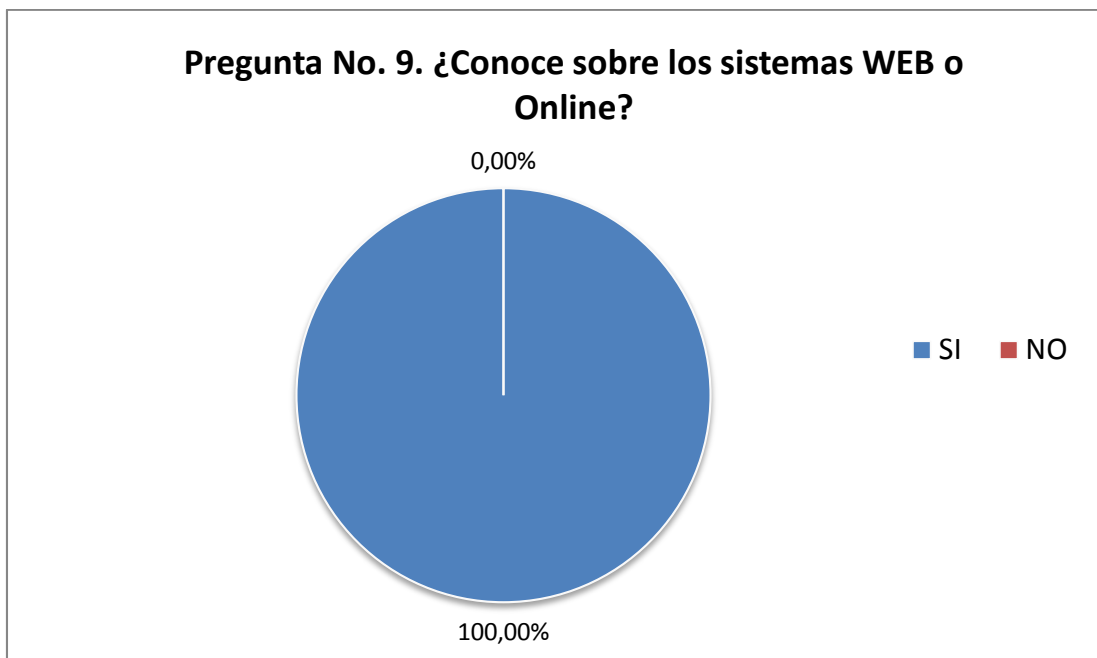


Fuente: Equipo de Investigación

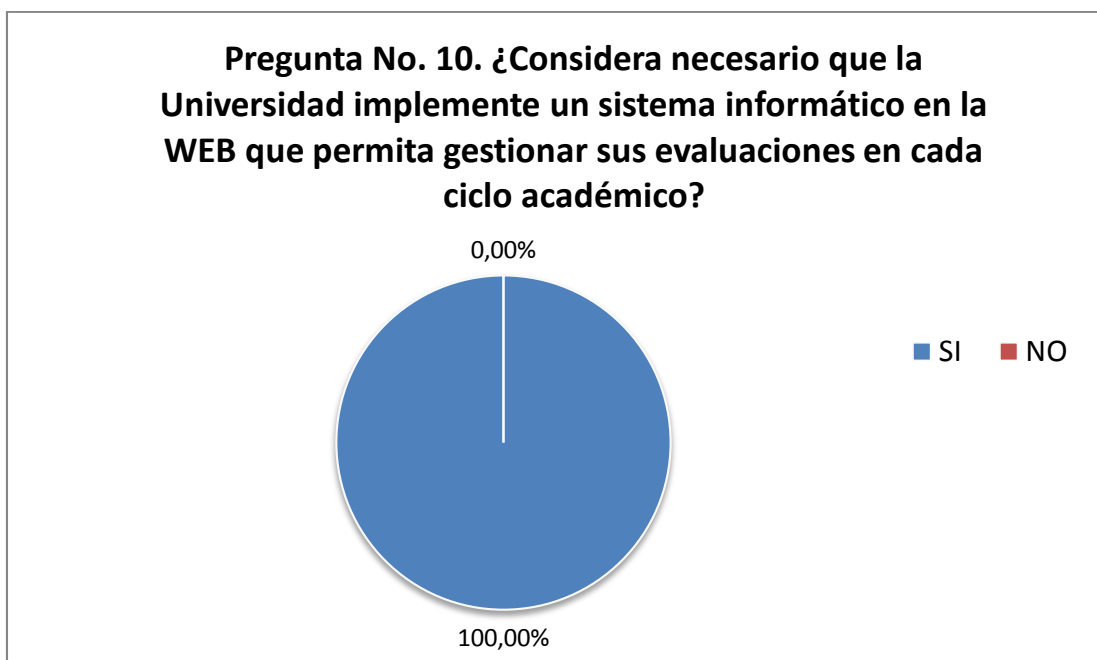
Pregunta No. 8. ¿Considera necesario que se realice un sistema informático que permita gestionar la evaluación del personal docente teniendo en cuenta las evaluaciones por los distintos canales?



Fuente: Equipo de Investigación



Fuente: Equipo de Investigación



Fuente: Equipo de Investigación